UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Predicción de incertidumbres en demandas mediante Procesos Gaussianos

Adrián Ramírez del Río
Tutor: Álvaro Barbero Jiménez
Ponente: Carlos Santa Cruz Fernández

ENERO 2015

Abstract

The inventory control problem aims to optimize the production and management of stocked products in order to meet their demand. A common added difficulty arises when the demand is uncertain since knowing the demand distribution (or an approximation to it) is essential so as to resolve the optimization problem. As a result, we would like to be able to predict the future demand and its distribution accurately. This uncertainty prediction problem which is an extended regression problem is the case of study of this document.

Wanting to know which techniques provide better results in solving this problem we study three regression models: Gaussian processes (*GPR*), *AdaBoost*(*ABR*) and *Support Vector Machines*(*SVR*). Each one of these models tackles the problem from a different perspective.

GPR models try to predict the demand distribution directly from data (Bayesian approach); *ABR* predicts the median of the distribution by combining the results of a set of regression trees and can be adjusted to predict any wanted quantile; meanwhile, *SVR* predicts the demand and a method valid for any regression model is applied to estimate the uncertainty on the demand.

Some experiments are conducted in a controlled environment, i. e., with an artificially generated dataset. In this data, the input is a real number and the output is a linear function of the input plus some random Gaussian noise. All three models are trained and tested with the generated data, finding in the results that *GPR* outperforms the other two models.

The next step is to experiment in a dataset extracted from a real inventory control problem, more concretely, the cash demand problem that bank branches suffer from. After analyzing the performance metrics achieved by each model on a total of 46 different datasets we conclude that again *GPR* is the model with the greater prediction capability.

Hence it is concluded that the Bayesian approach followed by *GPR* in order to solve the inventory control problem gives substantial advantages versus classical machine learning approaches.

Keywords: inventory control, regression, Gaussian processes, AdaBoost, SVR, uncertainty prediction, quantile estimation.

Resumen

El problema del control del inventario, que consiste en tratar de optimizar la producción y gestión de artículos para cubrir una demanda de los mismos, presenta en ocasiones la dificultad adicional de contener incertidumbres en la demanda, ya que conocer la distribución de las demandas (o una aproximación) es esencial para poder resolver el problema de optimización. Debido a esto, nos interesa ser capaces de predecir las demandas futuras y su distribución. Este problema de predicción de incertidumbres en demandas, que no es más que un caso extendido del problema de regresión, es el estudiado en este trabajo.

Con la intención de ver qué técnicas funcionan mejor en la resolución de este problema, se estudian tres modelos de regresión: procesos gaussianos(GPR), *AdaBoost*(ABR) y *Support Vector Machines*(SVR). Cada uno de estos modelos enfrenta el problema desde un punto de vista distinto.
Los modelos GPR intentan predecir directamente la distribución de las demandas (enfoque bayesiano); ABR predice la mediana de la distribución combinando árboles de regresión, pero puede ajustarse para predecir el cuantil deseado; por su parte, SVR realiza una predicción de la demanda y mediante un método general (aplicable a cualquier modelo de regresión) obtenemos predicciones de las incertidumbres en la demanda.

A continuación se realizan pruebas de los modelos en un entorno controlado, es decir, con un conjunto de datos generados artificialmente. En este conjunto la entrada es un número real y la salida una función lineal del mismo a la que se le suma ruido gaussiano. Los tres modelos se entrenan y evalúan con estos datos, hallando que GPR supera en rendimiento a los otros dos.
El siguiente paso es realizar experimentos en un conjunto de datos extraído de un problema del control del inventario real. Éste es el problema de la demanda de efectivo en sucursales bancarias. Tras analizar las métricas de rendimiento obtenidas por cada modelo en un total de 46 conjuntos de datos distintos concluimos que nuevamente es GPR el modelo con mayor capacidad de predicción.

Así, se concluye que el enfoque bayesiano proporcionado por GPR para resolver el problema descrito del control del inventario proporciona ventajas considerables frente a los enfoques clásicos de aprendizaje automático.

Palabras clave: control del inventario, regresión, procesos gaussianos, AdaBoost, SVR, predicción de incertidumbres, estimación de cuantiles.

Índice general

Índice de tablas

Índice de figuras

1 Introducción

En este capítulo se explican las causas que motivan la realización de este trabajo. Se describen después los objetivos generales y específicos perseguidos. Finalmente, se expone la estructura completa del documento.

1.1. Motivación

Un desafío recurrente en el mundo de la logística es el problema del control del inventario (*inventory control problem*). Este problema consiste en tratar de optimizar la producción y gestión de una serie de artículos con los que se debe cubrir una cierta demanda, de forma que se logren dos objetivos:

- Garantizar que en cada día de operación se tienen disponibles artículos suficientes como para cubrir las demandas del día.
- Minimizar los costes que supone mantener artículos en *stock* de un día para otro, así como los costes de producción de estos artículos.

En su forma más simple, el problema se suele formular matemáticamente de la sigueinte manera:
Una tienda tiene x_k artículos en *stock* en el instante de tiempo k. Seguidamente, realiza una orden y recibe u_k artículos, vendiendo w_k, donde w sigue una distribución de probabilidad conocida. Esto es:

$$x_{k+1} = x_k + u_k - w_k, \tag{1.1}$$
$$u_k \geq 0. \tag{1.2}$$

La tienda tiene costes c_k, que se calculan en función del número de artículos almacenados y el número de artículos ordenados:

$$c_k = c(x_k, u_k) \tag{1.3}$$

y que normalmente se considerarán de forma aditiva:

$$c_k = g(x_k) + h(u_k). \tag{1.4}$$

Así, la solución al problema que busca la tienda es encontrar u_k de forma óptima, es decir:

$$\min_{u_k} \sum_{k=0}^{\infty} c_k. \tag{1.5}$$

En el caso descrito, la distribución de las demandas w_k es conocida y el problema puede resolverse utilizando técnicas clásicas de optimización lineal, programación dinámica u otras estrategias de teoría de inventariado [1]. Si además de esto los costes son independientes del tiempo k el problema resulta trivial, siendo la estrategia óptima el ordenar cada día las unidades exactas que van a demandarse. No obstante los problemas de inventariado reales incluyen restricciones mucho más complejas, como pueden ser:

- Límites a la cantidad de artículos que pueden mantenerse en el inventario, que pueden ser variantes en el tiempo.
- Límites a la capacidad de producción y por tanto de órdenes diarias, también posiblemente variantes en el tiempo.
- Retrasos entre el momento en el que se ordena la producción y el momento en el que se reciben los artículos en el punto de venta.
- Múltiples puntos de venta con costes de almacenamiento diferentes.
- Múltiples puntos de producción con costes diferentes por unidad producida.
- Limitaciones en los transportes entre puntos de producción y puntos de venta.
- Incertidumbre en las demandas (se desconoce la distribución de w).

Ésta última complicación es, de entre todas las listadas, la que mayor dificultad añade al problema, dado que el desconocer las demandas imposibilita tomar decisiones respecto a la cantidad de artículos que deberían mantenerse en inventario. Esta limitación se suele atajar asumiendo que las demandas siguen una determinada distribución de probabilidad, aplicando entonces técnicas más avanzadas como optimización robusta[1], o estocástica[2]. El problema, sin embargo, se relega en identificar una distribución de probabilidad adecuada a la realidad del problema.

[1] http://en.wikipedia.org/wiki/Robust_optimization

[2] http://en.wikipedia.org/wiki/Stochastic_programming

En el día a día de la sociedad actual, innumerables sectores deben hacer frente a las dificultades descritas. Un caso particular es la demanda de efectivo en sucursales bancarias, que presenta el inconveniente mencionado de que la demanda se desconoce. Otra modificación característica de este caso particular es que la demanda puede ser negativa (ingreso de efectivo en la sucursal). Así, surge la motivación por parte del sector bancario de abordar este problema, proporcionando datos de sucursales reales que han sido utilizados en los experimentos aquí presentados.

1.2. Objetivos

En este trabajo se explora la posibilidad de estimar las demandas futuras mediante el uso de un modelo de predicción, de forma que se puedan obtener estimaciones fiables de las mismas y así facilitar la consiguiente labor de optimización. La predicción de demandas futuras no es sino un problema de predicción de series temporales, el cual ha sido ya muy estudiado en la literatura y para el que se han propuesto diversidad de modelos. No obstante, cabe destacar que en el caso del problema del control del inventario no solo es necesario conocer las demandas más probables que acontecerán durante los próximos días, sino también qué incertidumbre existe en las mismas. Dicho de otra forma, se hace necesario disponer de estimaciones de la distribución de probabilidad de las demandas futuras para que pueda realizarse una optimización adecuada.
Con el fin de estimar la distribución de las demandas se consideran tres alternativas para abordar el problema:

- Utilizar un modelo de predicción para estimar la media esperada de la distribución y a partir de la salida del modelo obtener estimaciones de los cuantiles. Para este enfoque pueden considerarse modelos clásicos del aprendizaje automático como *Support Vector Regression (SVR)* o redes neuronales.

- Utilizar un modelo de predicción capaz de estimar directamente los cuantiles de la distribución. Esta perspectiva se obtiene con modelos más específicos como por ejemplo la versión para regresión del algoritmo basado en árboles *AdaBoost* o modelos que realicen regresión cuantílica (*Quantile Regression*).

- Utilizar un modelo cuya predicción es en sí una distribución de probabilidad. Este planteamiento nos lo proporcionan los métodos de inferencia bayesianos, cuyo objetivo es encontrar una distribución a posterioi a partir de los datos.

Este trabajo explora los tres enfoques y los modelos elegidos para cada uno han sido, respectivamente, *SVR*, *AdaBoost* para regresión (*ABR*) y procesos

gaussianos para regresión (*GPR*). No obstante, nos centraremos principalmente en el estudio de éste último ya que proporciona una perspectiva más general y que consideramos más adecuada para el problema.
El objetivo de éste trabajo ha sido, por tanto, el estudio del marco teórico de *GPR* así como su aplicación, para, posteriormente, realizar una comparativa entre los tres modelos considerados, midiendo el rendimiento en un conjunto de datos generados artificialmente en el que se conoce la distribución de los mismos y en un conjunto de datos reales de sucursales bancarias interpretando los resultados para intentar determinar qué modelo es mejor en este contexto.
Cabe destacar también que una solución completa al problema del control del inventario requiere de muchos componentes, entre los cuales se incluyen la elección de una representación apropiada del problema en forma de programa de optimización, la elección de una estrategia de optimización acorde a esa representación, o la implementación de un algoritmo de optimización. De entre todos estos componentes este trabajo se centra únicamente en la previsión de las demandas futuras y su distribución, lo cual ya supone un problema de suficiente entidad para justificar un estudio en profundidad.

1.3. Estructura del documento

En las distintas secciones expuestas a lo largo de este documento, se presentan las distintas fases que se han adoptado para afrontar el problema de aprendizaje automático descrito en esta Introducción.
Así, el capítulo 2 comienza definiendo formalmente el problema de predicción al que nos enfrentamos. Seguidamente encontramos los conceptos teóricos que definen los procesos gaussianos, que son el campo de estudio principal de este trabajo. También aquí se definen brevemente los modelos *ABR* y *SVR*.
El capítulo 3 se inicia enumerando las tecnologías que han sido utilizadas para llevar a cabo los experimentos. A continuación se describe el proceso mediante el que se han construido los conjuntos de datos artificiales. También se introducen una serie de métricas que son utilizadas, tras presentar los resultados de cada modelo, para realizar comparaciones y extraer conclusiones.
Los experimentos con conjuntos de datos reales se recogen en el capítulo 4. En primer lugar se describen las características de los conjuntos de datos con los que se realizan los experimentos. Posteriormente, se definen una serie de métricas que nos permitirán medir el rendimiento de los modelos. Entonces, se describe la metodología seguida en los experimentos para cada modelo. El capítulo acaba con una comparativa de los resultados obtenidos.
Para terminar, presentamos en el capítulo 5 las conclusiones extraídas del trabajo realizado y mercamos las líneas de trabajo a seguir en el futuro.

2 Predicción de incertidumbres

Este capítulo presenta los conceptos teóricos en los que se basan los experimentos descritos en los capítulos posteriores. Así, en la sección 2.1 se explica cómo interpretar matemáticamente un problema de predicción. El siguiente apartado 2.2 expone todos los conceptos, métodos e interpretaciones necesarios para comprender cómo se aplican los procesos gaussianos para resolver un problema de predicción. En los puntos 2.3 y 2.4 se describen de forma breve los conceptos relacionados con los modelos *ABR* y *SVR*.

2.1. Presentación del problema

Como se ha comentado en la Introducción, el problema del control del inventario nos presenta la necesidad de ser capaces de predecir la distribución de la demanda cuando ésta se desconoce.
En esta sección se divide el problema de predicción de la distribución en dos partes: el problema de regresión clásico y el problema de estimación de cuantiles.

2.1.1. El problema de regresión

En el campo del aprendizaje automático, el aprendizaje supervisado consiste en inferir una función, mediante un proceso denominado entrenamiento, a partir de un conjunto de datos constituido por ejemplos de entrenamiento (también llamados patrones u observaciones) [2]. Cada patrón de entrenamiento está formado por una entrada (normalmente un vector) y un valor de salida o respuesta (un número o una etiqueta).

Dependiendo del tipo de los valores de salida, el problema del aprendizaje supervisado se divide en dos categorías: regresión y clasificación. Cuando la salida es discreta, se trata de un problema de clasificación. Por el contrario, si la salida es continua (el caso estudiado en este trabajo) nos enfrentamos a un problema de regresión.
Los modelos de regresión utilizan la función inferida como estimador de los

valores de salida dada una entrada.

Definido formalmente, partimos de un conjunto de datos de entrenamiento:

$$\begin{aligned} D &= \{(\boldsymbol{x}_i, y_i) : \boldsymbol{x}_i \in \mathcal{X},\ y_i \in \mathbb{R}\}_{i=1}^{N} \\ &= (\boldsymbol{X}, \boldsymbol{y}) \end{aligned} \tag{2.1}$$

dónde $\mathcal{X}$ es el espacio de características, los $\boldsymbol{x}_i$ se denominan *vectores de características* y las y_i *respuestas*.
Asumimos que existen f y ε tal que

$$y_i \ = \ f(\boldsymbol{x}_i) + \varepsilon \ = \ \mathrm{f}_i + \varepsilon \tag{2.2}$$

siendo $f : \mathcal{X} \longrightarrow \mathbb{R}$ la función subyacente a los datos y ε el posible ruido aleatorio presente en los datos. El objetivo es construir un estimador para la función subyacente

$$\bar{f} \approx f$$

de manera que dado cualquier *vector de características* $\boldsymbol{x}_* \in \mathcal{X}$ podemos predecir el valor de la *respuesta* y_* como $\bar{f}(\boldsymbol{x}_*)$.
No obstante, además de obtener predicciones para los valores de *respuesta*, también nos gustaría proporcionar una estimación del error que el modelo ha cometido en sus estimaciones, es decir, predecir la incertidumbre. De esta manera se puede construir un intervalo de confianza para las predicciones. Al problema de predicción de estas incertidumbres lo denominaremos *problema de estimación de cuantiles*.

2.1.2. El problema de estimación de cuantiles

Con el objetivo de construir los intervalos de confianza mencionados en el apartado 2.1.1, queremos ser capaces de estimar cuantiles de la distribución de la variable de respuesta. Para ello buscamos una función $\bar{q}_\alpha$ que aproxime el α-cuantil a partir de las predicciones $\boldsymbol{y}_*$.

Este objetivo es el mismo que aquel perseguido por la *regresión de cuantiles*[3] cuyo objetivo es estimar la mediana condicional (u otro cuantil) de la variable de respuesta y a partir de los valores observados $\boldsymbol{x}$, en contraposición a los algoritmos de regresión clásicos que se basan en buscar estimadores para la media condicional.
No obstante, nuestro objetivo es realizar ambas tareas: en lugar de entrenar un modelo para estimar directamente un α-cuantil o la media, estamos interesados en entrenar un modelo que realice predicciones de la media condicional de la respuesta y que *a la vez* permita extraer predicciones para los cuantiles.

2.2. Regresión mediante Procesos Gaussianos (GPR)

2.2.1. Conceptos previos

En primer lugar, puede hallarse a continuación un conjunto de definiciones de conceptos probabilísticos que deben conocerse para entender la base teórica de los procesos gaussianos así como su aplicación al problema de regresión.

Definición 1 *(Variable aleatoria) Sea Ω el espacio muestral asociado a un experimento, una variable aleatoria real X es una función real definida en Ω:*

$$X : \Omega \longrightarrow A \subseteq \mathbb{R} \tag{2.3}$$

donde A representa el conjunto de posibles valores que puede tomar X. Un vector $\boldsymbol{X} = (X_1, ..., X_n)$ formado por las variables aleatorias $X_1, ..., X_n$ se denomina vector aleatorio.

En general, se diferencian dos tipos de variables aleatorias. Si los valores que puede tomar X están restringidos a un conjunto discreto (numerable), se denomina variable discreta. Por el contrario, si el conjunto de valores posibles de X es no numerable (por ejemplo, un intervalo de los números reales), se dice que es continua.

Definición 2 *(Función de distribución) Para cualquier $x \in \mathbb{R}$, la función de distribución de una variable aleatoria X viene dada por:*

$$F_X(x) = \mathbb{P}(X \leq x). \tag{2.4}$$

Es decir, la probabilidad de que la variable aleatoria X sea menor o igual que el valor x considerado.
Análogamente, para un vector aleatorio $\boldsymbol{X}$, la función de distribución conjunta viene dada por:

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = F_{\boldsymbol{X}}(x_1, ..., x_n) = \mathbb{P}(X_1 \leq x_1, ..., X_n \leq x_n). \tag{2.5}$$

Definición 3 *(Función de densidad de probabilidad) La función de densidad de probabilidad (o simplemente función de densidad) de una variable aleatoria X continua es la derivada de su función de distribución F_X:*

$$f_X(x) = \frac{dF_X(x)}{dx} \tag{2.6}$$

La función de densidad conjunta de un vector aleatorio $\boldsymbol{X}$ *es:*

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{\partial^n F_{\boldsymbol{X}}(x_1, ..., x_n)}{\partial x_1 ... \partial x_n} \tag{2.7}$$

En general, las variables aleatorias se expresan a través de su función de densidad.

Otra función relacionada con la función de distribución de una variable aleatoria X en la que estamos interesados es la función cuantil, que se define como sigue:

Definición 4 *(Función cuantil) Sean* X *una variable aleatoria con distribución* F_X *y* $\alpha \in (0,1) \subset \mathbb{R}$*. Definimos la siguiente función:*

$$Q_X(\alpha) = \inf_{x \in \mathbb{R}} \{\alpha \leq F_X(x)\} \tag{2.8}$$

Al valor $q_\alpha = Q_X(\alpha)$ *lo llamamos el* α*-cuantil de* F_X*.*

Se trata de una definición alternativa para la inversa de la función de distribución F_X^{-1}, también denominada *percent point function* en la literatura en inglés. La función cuantil dada una distribución arbitraria no es fácil de hallar, ya que en muchas ocasiones se desconoce una forma cerrada de la misma. No obstante, las principales distribuciones clásicas permiten el cálculo directo o aproximaciones muy precisas [4].
Seguidamente, se expone uno de los conceptos clave en el que se basan los procesos gaussianos, la función de distribución normal (o gaussiana) para un vector aleatorio.

Definición 5 *(Distribución normal multivariante) Se dice que un vector aleatorio* $\boldsymbol{X}$ *sigue una distribución normal multivariante si tiene la siguiente función de densidad de probabilidad conjunta*

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right) \tag{2.9}$$

El vector $\boldsymbol{\mu}$ *representa la esperanza matemática (o media) de* $\boldsymbol{X}$ *y la matriz* $\Sigma = AA^T$ *es la matriz de covarianza de las componentes* X_i*.*

A continuación, se presenta una propiedad muy importante que cumplen los vectores aleatorios con distribución normal multivariante.

Definición 6 *(Propiedad de marginalización) Se trata de la siguiente propiedad:*

$$\boldsymbol{X}_1 = (x_1, \ldots, x_n) \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}),$$
$$\boldsymbol{X}_2 = (x'_1, \ldots, x'_m) \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22})$$

si y sólo si

$$\begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \right).$$

Es decir, que dados dos vectores aleatorios X_1 y X_2 con distribución normal, examinar el conjunto de variables formado por ambos sigue también una distribución normal que depende de los parámetros de las distribuciones de los vectores y de la correlación entre ambos (Σ_{12} y Σ_{21}). Esto significa que al examinar un conjunto de variables mayor no cambia la distribución del conjunto examinado hasta ese momento.

Finalmente, se presenta un teorema que es la base de la aplicación de los procesos gaussianos:

Teorema 1

Dados dos vectores aleatorios independientes $\boldsymbol{a}$ *y* $\boldsymbol{b}$ *tal que*

$$\begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \begin{bmatrix} A & C^\top \\ C & B \end{bmatrix} \right)$$

se cumple

$$\boldsymbol{b}|\boldsymbol{a} \sim \mathcal{N}\left(\boldsymbol{\mu}_b + CA^{-1}(\boldsymbol{a} - \boldsymbol{\mu}_a),\ B - CA^{-1}C^\top\right)$$

es decir,

$$\mathbb{E}[\boldsymbol{b}|\boldsymbol{a}] = \boldsymbol{\mu}_b + CA^{-1}(\boldsymbol{a} - \boldsymbol{\mu}_a),$$
$$\mathbb{V}[\boldsymbol{b}|\boldsymbol{a}] = B - CA^{-1}C^\top.$$

Este teorema se puede interpretar del siguiente modo: conociendo la distribución de un conjunto de variables aleatorias normales, uno puede dividirlo en dos subconjuntos (a y b) y si uno sabe qué valores toma uno de ellos (a), se puede calcular la distribución del otro (b) condicionada a esos valores observados. Como veremos más adelante, al hablar de procesos gaussianos esto se reduce a que el conjunto de patrones de entrenamiento son valores observados de a, y a partir de éstos calculamos la distribución de los patrones de test, que serían b.

2.2.2. Los procesos gaussianos

Con la introducción de los conceptos básicos de la sección anterior, esta sección se centra en los aspectos teóricos de los procesos gaussianos, que son el área principal de estudio del presente trabajo. Los procesos gaussianos tienen su base en estadística y aprendizaje automático, y son considerados una herramienta potente que se basa en un enfoque bayesiano frente al enfoque clásico en el que se minimiza una función de pérdida. Además, están relacionados con una variedad de modelos como *Support Vector Machines(SVM)*, modelos *Spline* y *Relevance Vector Machines* entre otros [5].
Se trata de un tipo particular de proceso estocástico ampliamente estudiado en el campo de la estadística que no es más que una generalización de la distribución de probabilidad gaussiana. Mientras que la distribución gaussiana se refiere a una única variable aleatoria, un proceso gaussiano está asociado a una colección de variables aleatorias y la distribución de subconjuntos de esta colección. La definición es la siguiente:

Definición 7 *(Proceso Gaussiano) Un proceso gaussiano consiste en una colección (potencialmente infinita) de variables aleatorias, cada una de las cuales sigue una distribución normal, para las que dado cualquier subconjunto finito de las mismas, éste sigue una distribución normal multivariante.*

Observar que por cómo se definen, éstos procesos cumplen la *propiedad de marginalización*, previamente presentada en el apartado (2.2.1).
Un proceso gaussiano lo denotamos como

$$f\left(\boldsymbol{x}\right) \sim \mathcal{GP}\Big(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\Big). \tag{2.10}$$

Esta notación significa que para cada valor $\boldsymbol{x} \in \mathbb{R}^n$, consideramos la función $f\left(\boldsymbol{x}\right)$ como una variable aleatoria, es decir, se define una distribución sobre el conjunto de funciones reales, siendo

$$m\left(\boldsymbol{x}\right) \stackrel{\text{def}}{=} \mathbb{E}\left[f\left(\boldsymbol{x}\right)\right], \tag{2.11}$$

la función de media y

$$k\left(\boldsymbol{x}, \boldsymbol{x}'\right) \stackrel{\text{def}}{=} \mathbb{E}\Big[\left(f\left(\boldsymbol{x}\right) - m\left(\boldsymbol{x}\right)\right)\left(f\left(\boldsymbol{x}'\right) - m\left(\boldsymbol{x}'\right)\right)\Big] \tag{2.12}$$

la función de covarianza.
Intuitivamente esto significa que el valor de $f\left(\boldsymbol{x}\right)$ se trata como una variable aleatoria con media $m\left(\boldsymbol{x}\right)$y con covarianza entre pares de variables aleatorias

$$cov\left(f\left(\boldsymbol{x}\right), f\left(\boldsymbol{x}'\right)\right) = k\left(\boldsymbol{x}, \boldsymbol{x}'\right) = k\left(\boldsymbol{x}', \boldsymbol{x}\right). \tag{2.13}$$

2.2.3. Notación

En esta sección presentamos una notación que nos servirá para explicar el funcionamiento de *GPR* de una manera más sencilla.
Dada una función de covarianza $k(\boldsymbol{x}, \boldsymbol{x}')$ y dos conjuntos $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \in \mathcal{X}^N$, $\boldsymbol{X}' = (\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_N) \in \mathcal{X}^M$ definimos la siguiente operación

$$k\left(\boldsymbol{X}, \boldsymbol{X}'\right) = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}'_1) & k(\boldsymbol{x}_1, \boldsymbol{x}'_2) & \ldots & k(\boldsymbol{x}_1, \boldsymbol{x}'_M) \\ k(\boldsymbol{x}_2, \boldsymbol{x}'_1) & k(\boldsymbol{x}_2, \boldsymbol{x}'_2) & \ldots & k(\boldsymbol{x}_2, \boldsymbol{x}'_M) \\ \vdots & \vdots & \vdots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}'_1) & k(\boldsymbol{x}_N, \boldsymbol{x}'_2) & \ldots & k(\boldsymbol{x}_N, \boldsymbol{x}'_M) \end{bmatrix} \tag{2.14}$$

Para un conjunto de entrenamiento $D = (\boldsymbol{X}, \boldsymbol{y})$ y un vector de características $\boldsymbol{x}_* \in \mathcal{X}$ del que queremos predecir la respuesta, introducimos la siguiente notación

$$K = k(\boldsymbol{X}, \boldsymbol{X}) \tag{2.15}$$
$$\boldsymbol{k}_* = k(\boldsymbol{X}, \boldsymbol{x}_*) \tag{2.16}$$
$$\boldsymbol{k}_{**} = k(\boldsymbol{x}_*, \boldsymbol{x}_*) \tag{2.17}$$

y análogamente para un conjunto de test $\boldsymbol{X}_* = \left(\boldsymbol{x}_*^1, \boldsymbol{x}_*^2, \ldots, \boldsymbol{x}_*^M\right) \in \mathcal{X}^M$, formado por varios vectores de características, escribimos

$$K_* = k(\boldsymbol{X}, \boldsymbol{X}_*) \tag{2.18}$$
$$K_{**} = k(\boldsymbol{X}_*, \boldsymbol{X}_*) \tag{2.19}$$

2.2.4. Predicción con Procesos Gaussianos

En esta sección se explica cómo los procesos gaussianos pueden utilizarse para resolver un problema de regresión. Se trata de un modelo no paramétrico, es decir, la fase de entrenamiento del modelo no consiste en la optimización de una serie de parámetros, si no que se trata simplemente de la inversión de una matriz (como veremos más adelante). Antes de analizar ningún dato, asumimos que la función subyacente f sigue un proceso gaussiano y que el ruido es gaussiano e independiente idénticamente distribuido:

$$f \sim \mathcal{GP}\Big(m(\boldsymbol{x}), k\left(\boldsymbol{x}, \boldsymbol{x}'\right)\Big); \tag{2.20}$$

$$\varepsilon \sim \mathcal{N}\left(0, \sigma^2\right). \tag{2.21}$$

De cara a simplificar la notación suele asumirse que el proceso tiene media cero [5], aunque no es necesario, obteniendo:

$$f(x) \sim \mathcal{GP}\Big(0, k\left(\boldsymbol{x}, \boldsymbol{x}'\right)\Big)$$

o expresado de otra forma:

$$f|\boldsymbol{X} \sim \mathcal{N}(0, K) \tag{2.22}$$

Visto desde el punto de vista bayesiano, estamos definiendo una distribución a priori sobre el espacio de funciones, la cual queda determinada únicamente por la elección de la función de covarianza. Es decir, dependiendo de nuestra elección de la función de covarianza, consideraremos que ciertas funciones tienen una probabilidad más alta de parecerse a la función subyacente que otras.
Sin embargo, esta elección debe llevarse a cabo de acuerdo a las características de los datos considerados en la medida de lo posible. En la sección 2.2.6 se estudian en más detalle este tipo de funciones.

Como ya se ha comentado, representamos las variables de respuesta en función de las características y el posible ruido presente en los datos del siguiente modo:

$$y_i \;=\; f(\boldsymbol{x}_i) + \varepsilon_i,\ \varepsilon_i \sim \mathcal{N}\left(0, \sigma^2\right) \tag{2.23}$$

y debido a que asumimos que las variables aleatorias f y ε siguen una distribución normal, la variable de respuesta $\boldsymbol{y}$ seguirá también una distribución normal. Puesto que las medias las distribuciones tanto de f como de ε son 0, la distribución de $\boldsymbol{y}$ tendrá también media 0.
Así pues, para conocer la distribución exacta de las respuestas $\boldsymbol{y}$ basta darse cuenta de que la covarianza entre dos muestras aleatorias de $\boldsymbol{y}$ puede expresarse de la siguiente forma:

$$\begin{aligned} cov(y_p, y_q) &= cov(f(\boldsymbol{x}_p) + \varepsilon_p, f(\boldsymbol{x}_q) + \varepsilon_q) \\ &= cov(f(\boldsymbol{x}_p) + \varepsilon_p, f(\boldsymbol{x}_q)) + cov(f(\boldsymbol{x}_p) + \varepsilon_p, \varepsilon_q) \\ &= cov(f(\boldsymbol{x}_p), f(\boldsymbol{x}_q)) + cov(\varepsilon_p, f(\boldsymbol{x}_q)) + cov(f(\boldsymbol{x}_p), \varepsilon_q) + cov(\varepsilon_p, \varepsilon_q) \\ &= k(\boldsymbol{x}_p, \boldsymbol{x}_q) + \sigma^2 \delta_{pq}, \end{aligned}$$

siendo δ_{pq} la función delta de Kronecker

$$\delta_{pq} = \begin{cases} 1 & : \quad p = q \\ 0 & : \quad p \neq q \end{cases} \tag{2.24}$$

Esto nos permite por tanto calcular la distribución de las respuestas $\boldsymbol{y}$ de los patrones de entrenamiento a partir de las características $\boldsymbol{x}$:

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{0}, K_{\boldsymbol{y}}) \quad \text{donde} \quad K_{\boldsymbol{y}} = K + \sigma^2 \boldsymbol{I}. \tag{2.25}$$

Este resultado es importante pues nos indica cómo se distribuyen los valores objetivo en el conjunto de entrenamiento.

Ahora, dado un conjunto de vectores de características $\boldsymbol{X}_*$ nos gustaría ser capaces de calcular la distribución de probabilidad de las variables de respuesta de ese conjunto (en principio desconocidas). En otras palabras, queremos hallar la distribución de las $\boldsymbol{y}_*$ que se corresponden con $\boldsymbol{X}_*$.

Para ello, basta notar que puesto que $\boldsymbol{y}$ e $\boldsymbol{y}_*$ son variables aleatorias que siguen una distribución gaussiana (por la hipótesis asumida de que el proceso es gaussiano), deben cumplir la *propiedad de marginalización*. Es decir, esta propiedad nos permite representar la distribución conjunta de las variables de respuesta:

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}_* \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} K_{\boldsymbol{y}} & K_* \\ K_*^\top & K_{\boldsymbol{y}*} \end{bmatrix}\right) \tag{2.26}$$

donde

$$K_{\boldsymbol{y}_*} = K_{**} + \sigma^2 \boldsymbol{I}. \tag{2.27}$$

Una vez hallada la forma de esta distribución conjunta lo único que nos falta para obtener la distribución de las respuestas $\boldsymbol{y}_*$ es marginalizar con respecto a las variables $\boldsymbol{y}$.
Para ello, nos servimos del Teorema 1 que nos permite hallar la distribución de las $\boldsymbol{y}_*$ a partir de los datos de entrenamiento D y los vectores de características $\boldsymbol{X}_*$:

$$\boldsymbol{y}_* | D, X_* \ \sim \ \mathcal{N}\Big(\bar{\boldsymbol{y}}_*, \ \mathbb{V}\left(\bar{\boldsymbol{y}}_*\right)\Big). \tag{2.28}$$

siendo

$$\bar{\boldsymbol{y}}_* \quad \stackrel{\text{def}}{=} \quad K_* K_{\boldsymbol{y}}^{-1} \boldsymbol{y}. \tag{2.29}$$

y

$$\begin{aligned} \mathbb{V}\left(\bar{\boldsymbol{y}}_*\right) &= K_{\boldsymbol{y}_*} - K_*^\top K_{\boldsymbol{y}}^{-1} K_* \\ &= K_{**} + \sigma^2 \boldsymbol{I} - K_*^\top K_{\boldsymbol{y}}^{-1} K_* \end{aligned} \tag{2.30}$$

Esto significa que hemos encontrado de forma cerrada (tenemos una fórmula) la distribución de las respuestas $\boldsymbol{y}_*$ de los vectores de características $\boldsymbol{X}_*$ que queremos predecir. Esta distribución se conoce como la distribución gaussiana a posteriori, y de acuerdo a [5], el valor $\bar{\boldsymbol{y}}_*$ es el mejor estimador para la variable $\boldsymbol{y}_*$. Por tanto, podemos llevar a cabo el proceso de predicción a través de la media de esta distribución (2.29) y obteniendo además un valor de incertidumbre en las predicciones dado por la ecuación (2.30).

2.2.5. Estimación de cuantiles con GPR

En el apartado anterior se presentaron las ecuaciones (2.29) y (2.30) que determinan la distribución gaussiana a posteriori. Puesto que se trata de

una distribución ampliamente conocida, es fácil obtener el valor de un cuantil de la misma mediante la *función cuantil* definida en (2.2.1). El propio modelo nos proporciona una estimación de la incertidumbre presente en las predicciones. Esta facilidad a la hora de medir las incertidumbres de forma casi inmediata fue un motivo importante para que el trabajo se centrara en el estudio de este tipo de procesos.

2.2.6. Funciones de covarianza

En la sección (2.2.2) se mostró la forma de aplicar los procesos gaussianos al problema de regresión. Asumiendo una función de media cero para el proceso, el foco del problema se centra en la elección de una función de covarianza $k(x, x')$, también denominada a veces *kernel*.
La covarianza entre un par de variables aleatorias $f(x)$ y $f(x')$ se calcula de acuerdo a la ecuación (2.12). Cuando se asume que el proceso tiene media cero, la función de covarianza se traduce en:

$$k\left(x, x'\right) = \mathbb{E}\left[f(x)f(x')\right] \tag{2.31}$$

Se trata una función de medida entre dos variables aleatorias conjuntamente distribuidas que determina cómo de relacionadas están. Es decir, calcula la correlación entre las dos variables.

Una función de covarianza se dice *estacionaria* si depende de la diferencia $x - x'$, es decir, si es invariante a traslaciones en el espacio de características. Si además, la función de covarianza depende únicamente de $|x - x'|$ (considerando la distancia euclídea) se dice que es isotrópica, esto es, invariante a traslaciones y rotaciones.

En el aprendizaje con procesos gaussianos, la función de covarianza tiene un papel protagonista, pues la precisión de las predicciones dependerá en gran medida del *kernel* seleccionado. Como apunta [5], la función seleccionada debe cumplir ciertas propiedades: debe ser semi-definida positiva y simétrica.

Algunas de las funciones de covarianza más comunes se exponen a continuación:

- Función cuadrática exponencial:
  Se trata, probablemente, de la función más ampliamente utilizada a la hora de elegir una función de covarianza:

$$k_{se}\left(x, x'\right) = \sigma_f^2 exp\left(-\frac{1}{2}\sum_{k=1}^{d}\frac{\left(x_{i,k} - x_{j,k}\right)^2}{l_k^2}\right) \tag{2.32}$$

El parámetro σ_f^2 expresa la variabilidad del proceso gaussiano mientras que l_k marca la correlación en la k-ésima dimensión del espacio de características. Se trata de una función estacionaria y anisotrópica (excepto si $l_k = l \; \forall \; k$, caso en el que es isotrópica).

- Función exponencial

$$k_{exp}\left(x, x'\right) = \sigma_f^2 exp\left(-\sqrt{\sum_{k=1}^{d} \frac{(x_{i,k} - x_{j,k})^2}{l_k^2}}\right) \tag{2.33}$$

La interpretación de los parámetros es la misma que en la función cuadrática exponencial.

- Función lineal:

$$k_{lin}\left(x, x'\right) = x_i^\top \Sigma x_j \tag{2.34}$$

La matriz diagonal $\Sigma = (\sigma_1, ..., \sigma_D)$ contiene las varianzas a priori de los coeficientes del modelo lineal.

Ya se ha hecho notar que la elección de la función de covarianza marca notablemente el resultado que se obtiene al utilizar los procesos gaussianos para regresión. Incluso una vez elegida una familia de funciones como las previamente presentadas, debemos elegir un valor para los parámetros del *kernel*. Este es un problema difícil, ya que el rango de valores posibles es muy amplio y el modelo es bastante sensible a cambios en los parámetros. Por este motivo, los parámetros de la función de covarianza se toman como los hiperparámteros libres del modelo que deben ajustarse.

2.2.7. Ajuste de los hiperparámetros

El conjunto de hiperparámetros del modelo, representado mediante un vector θ, está formado por los parámetros del *kernel* y el valor σ^2 en 2.23 (la varianza a priori del error en los datos). Por ejemplo, para la función cuadrática exponencial,

$$\theta = \left\{l, \sigma_f^2, \sigma^2\right\} \tag{2.35}$$

El modelo *GPR* analiza el conjunto de datos de entrenamiento y aplicando técnicas de modelado bayesiano, infiere un valor para los mismos que proporcione una buena capacidad de generalización. Para ello aplica el método de *maximización de la verosimilitud marginal (marginal likelihood maximization)* también conocido en el contexto de los procesos gaussianos como *MAP estimate*(estimador del máximo a posteriori, o *Maximum A Posteriori* en inglés).
De acuerdo al teorema de Bayes, podemos representar la probabilidad a posteriori de los hiperparámetros como:

$$\mathbb{P}\left(\theta|\boldsymbol{X}, \boldsymbol{y}\right) = \frac{\mathbb{P}(\boldsymbol{y}|\boldsymbol{X}, \theta)\mathbb{P}(\theta)}{\mathbb{P}(\boldsymbol{y}|\boldsymbol{X})} \tag{2.36}$$

$\mathbb{P}(\boldsymbol{y}|\boldsymbol{X},\theta)$ es la *verosimilitud marginal* que debe maximizarse y $\mathbb{P}(\theta)$ es la probabilidad a priori de los hiperparámetros (la elección más general es considerar que todos los posibles valores tienen la misma probabilidad, es decir, una distribución uniforme a lo largo de todo el rango de posibles valors).
La verosimilitud marginal puede calcularse a través de la siguiente integral [5]:

$$\mathbb{P}\left(\boldsymbol{y}|\boldsymbol{X},\theta\right) = \int \mathbb{P}\left(\boldsymbol{y}|f,\boldsymbol{X},\theta\right)\mathbb{P}\left(f|\boldsymbol{X},\theta\right)df \tag{2.37}$$

Puesto que estamos en el marco de los procesos gaussianos, aplicando las hipótesis asumidas en 2.2.4 ("todo"sigue una distribución normal) llegamos a la conclusión de que $\boldsymbol{y} \sim \mathcal{N}\left(0, K_y\right)$, y con esto podemos calcular el logaritmo de la integral 2.37 analíticamente, obteniendo:

$$log\mathbb{P}\left(\boldsymbol{y}|\boldsymbol{X},\theta\right) = -\frac{1}{2}\boldsymbol{y}\top K_y\boldsymbol{y} - \frac{1}{2}log\left|K_y\right| - \frac{n}{2}log2\pi \tag{2.38}$$

El primer término representa cómo de bien se ajusta el modelo a los datos, el segundo expresa la complejidad del modelo y el tercero es una constante de normalización. Así pues, encontrar los parámetros θ que máximizan esta función es encontrar los parámetros que están en mayor concordancia con los datos observados.

En resumen, el problema de ajuste de los hiperparámetros queda reducido a la maximización de la función $L(\theta) = log\mathbb{P}\left(\boldsymbol{y}|\boldsymbol{X},\theta\right)$ (o la minimización de -$L(\theta)$), que generalmente se resuelve aplicando un algoritmo de descenso por gradiente.

2.2.8. Apuntes finales

En este capítulo, se ha explicado el funcionamiento de GPR en su forma más simple. No obstante, tal y como indican Williams y Rasmussen [5], una serie de modificaciones se pueden añadir al modelo. Por ejemplo, no tiene por qué considerarse que el posible ruido en los datos es gaussiano, en realidad, podemos asumir cualquier distribución de probabilidad. No obstante, de elegir ruido no gaussiano, ya no es aplicable el método de ajuste de los hiperparámetros descrito en (2.2.7) y se deben adoptar técnicas más complejas para aproximación de integrales (*Grid Integration*, *Monte Carlo Integration*, o *Central composite design* son algunas de ellas).
También podríamos considerar que el proceso gaussiano no sigue una media cero, sino que sigue una función de media determinada. O podríamos definir distribuciones de probabilidad a priori sobre los posibles valores de los hiperparámetros. Esto nuevamente introduce complicaciones en la teoría y aplicación de *GPR*.

Así vemos que *GPR* se basa en una teoría que tiene mucha flexibilidad de modelado a través de la función de covarianza, las distribuciones a priori, etc. Esto es un arma de doble filo, ya que en la práctica estos modelos pueden resultar muy difíciles de entrenar y configurar adecuadamente.

2.3. Regresión mediante *AdaBoost (ABR)*

El nombre del modelo de aprendizaje automático *AdaBoost*, presentado inicialmente por Freund y Schapire [6] en 1997, proviene del inglés *Adaptive Boosting*, y es aplicable tanto a problemas de clasificación como de regresión. Más tarde ese mismo año, Drucker [7] modificó el algoritmo *AdaBoost* original para regresión *AdaBoost.R*, dando lugar al algoritmo *AdaBoost.R2*, explicado más en detalle en la sección (2.3.2).
Como su nombre indica, se trata de un meta-algoritmo de aprendizaje basado en el *Boosting*. Esto quiere decir que, a partir de una serie de modelos denominados *weak learners* (algoritmos de aprendizaje con mucho sesgo), el algoritmo *AdaBoost* combina sus predicciones mediante una media ponderada, obteniendo así la predicción final. De esta manera, se consigue una mejora en el rendimiento del modelo gracias a la reducción del sesgo.
El término *Adaptive* se refiere a que el entrenamiento de los *weak learners* es adaptativo, es decir, la forma de entrenar uno de éstos modelos débiles depende del resultado de los entrenados previamente.

2.3.1. Árboles de regresión *CART*

A la hora de elegir los denominados *weak learners* la comunidad del aprendizaje automático se decanta por los *árboles de decisión*, debido a su buen rendimiento [8]. Existen diferentes métodos de aprendizaje automático basados en árboles, como el algoritmo *CART(Classification And Regression Tree)* [9] o el algoritmo *C4.5* [10]. En esta sección se da una breve explicación del funcionamiento del método *CART*: una técnica no paramétrica de aprendizaje basado en árboles, que proporciona árboles de clasificación o de regresión, dependiendo del tipo (categórico o numérico) de la variable de respuesta.
Los árboles de regresión dividen el espacio de características ($\mathcal{X}$) recursivamente en regiones disjuntas a partir de un conjunto de reglas de partición. Estas reglas son simplemente comparaciones entre una característica (una dimensión de X) y un valor. El funcionamiento del algoritmo *CART* es el siguiente:
En cada nodo del árbol, se selecciona la regla que proporciona la "mejor"división binaria para diferenciar los patrones de entrenamiento con respecto a esa característica. El criterio para decidir qué regla proporciona una mejor división se mide mediante la reducción de alguna función de pérdida

(generalmente la suma de los errores cuadráticos).
Una vez que una regla ha sido seleccionada y se ha dividido un nodo en otros dos, se lleva a cabo el mismo proceso en cada nodo hijo.
Para decidir cuándo detener el proceso de división existen dos estrategias: parar cuando unos criterios preestablecidos se satisfacen (pre-poda); o construir los árboles completos (cada nodo hoja contiene una observación) y aplicar después un algoritmo de poda.
Observar que cada rama del árbol termina en un nodo hoja, que se corresponde con una región disjunta del espacio X. Por ello, cada observación cae dentro de una única región, y cada región queda definida por un único conjunto de reglas.
De este modo, a la hora de realizar predicciones sobre un patrón, se le aplican las reglas de división del árbol entrenado, identificando así la región a la que éste pertenece, y se devuelve como predicción la media de los valores de respuesta de los patrones de entrenamiento que pertenecen a esa región.

2.3.2. El algoritmo *AdaBoost.R2*

En la aplicación del algoritmo *AdaBoost.R2* [7] se utilizan como *weak learners* los árboles de regresión descritos en (2.3.1).
Durante el entrenamiento, éstos se entrenan de manera secuencial. El primer árbol se entrena seleccionando N muestras aleatorias con reemplazamiento del conjunto de observaciones $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ y considerando todas equiprobables. Después, se evalúan todos los patrones en D, y según el error de predicción cometido en cada patrón se ajusta la probabilidad de seleccionarlo para el entrenamiento del siguiente árbol, de modo que sea más probable escoger los patrones con mayor error, es decir, más difíciles de predecir. Así, conforme más árboles construimos, más difíciles son los conjuntos de entrenamiento. Además, los distintos árboles se especializan en distintas partes del espacio de características, ya que los conjuntos con los que se entrenan son diferentes.
Una vez se han entrenado todos los árboles, se combinan mediante la mediana ponderada, teniendo en cuenta que aquellos que estén más seguros de sus predicciones tendrán más importancia.

2.3.3. Estimación de cuantiles con *ABR*

Como ya se ha descrito previamente, el algoritmo *AdaBoost.R2* predice directamente la mediana de la distribución de la variable de respuesta. Para ello, calcula la mediana ponderada de las predicciones de los *weak learners*. Así, una forma muy simple de estimar un α-cuantil con este modelo es obtener el correspondiente α-cuantil ponderado de las predicciones de los árboles. Ésta es, precisamente, la aproximación que se sigue en este trabajo a la hora

de estimar los cuantiles de la distribución de la variable de respuesta con el modelo *ABR*.

2.4. Regresión mediante *SVM (SVR)*

Las máquinas de vectores de soporte o *Support Vector Machines (SVM)* son modelos paramétricos de aprendizaje supervisado utilizados para análisis de regresión y clasificación. La teoría fue inicialmente desarrollada para problemas de clasificación por Vapnik y su equipo de trabajo en los laboratorios de $AT\&T$ [11] [12], y ha sido ampliamente difundida en el campo del aprendizaje automático gracias a Schölkopf y Smola [13]. Una versión de *SVM* para regresión, conocida como *Support Vector Regression (SVR)*, fue propuesta poco después [14].

2.4.1. *Support Vector Machines*

La motivación detrás de la aparición de este modelo es resolver el problema de clasificación. Supongamos que tenemos un conjunto de patrones, cada uno perteneciente a una de dos clases, siendo el objetivo decidir a qué clase pertenece un nuevo patrón. El modelo interpreta los patrones como un vector n-dimensional en el espacio de características, y queremos saber si podemos separar todos los puntos (patrones) con un hiperplano (n-1)-dimensional, y en tal caso encontrar el mejor. Esto se conoce como un clasificador lineal. Pueden existir muchos de estos hiperplanos, pero una elección razonable es aquel que proporciona un margen mayor entre las dos clases. Así se elige el hiperplano cuya distancia al punto más cercano de cada clase se maximiza, que en general proporciona el error de generalización menor.

Aunque el problema original se centra en un espacio de dimensión finita, ocurre a menudo que las clases no son separables linealmente en el espacio original por lo que se propuso transformarlo a través de una función Φ en uno de dimensión mucho más alta, con el objetivo de hacer la separación más fácil. Las transformaciones usadas en *SVM* se diseñan de modo que el producto escalar sea fácilmente computable a partir de las variables del espacio original. Para ello, se definen con respecto a una función *kernel*:

$$k\left(x, x'\right) = \Phi\left(x\right) \cdot \Phi\left(x'\right) \tag{2.39}$$

La búsqueda de este hiperplano puede reescribirse como un problema de optimización, el cual puede modificarse para resolver el problema de regresión, obteniendo así el modelo *SVR*.

2.4.2. Estimación de cuantiles con SVR

Puesto que *SVR* no es un algoritmo que proporcione un nivel de incertidumbre de sus predicciones y no está pensado para conseguir estimaciones de los cuantiles, se diseñó el método explicado a continuación para abordar este problema.

Inicialmente, se divide el conjunto de entrenamiento $D = (\boldsymbol{x}_i, y_i)_{i=1}^N$ en dos conjuntos disjuntos

$$D_1 = (\boldsymbol{x}_i, y_i) \; \forall \, i \in \left(1, \ldots, \frac{N}{2}\right) \quad \text{y} \tag{2.40}$$

$$D_2 = (\boldsymbol{x}_i, y_i) \; \forall \, i \in \left(\frac{N}{2} + 1, \ldots, N\right), \tag{2.41}$$

de manera que la mitad de los patrones quedan en D_1 y la otra mitad en D_2. A continuación, se entrena el modelo tomando como conjunto de entrenamiento D_1, y se aplica al conjunto D_2, obteniendo así las predicciones $\bar{y}_{\frac{N}{2}+1}, ..., \bar{y}_N$.

Ahora, para cada una de éstas predicciones se calcula su error $e_i = \bar{y}_i - y_i$, obteniendo el conjunto de errores $E = \{e_{\frac{N}{2}+1}, ..., e_N\}$ que es guardado junto con el modelo.
Dado un conjunto de test $D_* = \{(\boldsymbol{x}_*^i, y_*^i)\}_{i=1}^M$ se obtienen las predicciones del modelo $y_*^1, ..., y_*^M$ y se extrae el α-cuantil muestral E_α del conjunto de errores E, para finalmente devolver como estimación del cuantil en el punto x_*^i el valor $\bar{q}_\alpha^i = y_*^i + E_\alpha$. Es decir, a cada predicción del modelo se le suma el α-cuantil muestral de los errores obtenidos en las predicciones del conjunto de validación D_2.
Observar que un inconveniente de éste método es que la banda de confianza generada al estimar un cuantil es constante, es decir, se está considerando que la incertidumbre en las predicciones es siempre la misma, lo cual no es necesariamente cierto.

3 Experimentos con datos artificiales

Este capítulo contiene la información referente a los resultados experimentales con datos artificiales obtenidos por los modelos presentados en 2.
En primer lugar, se describen en la sección 3.1 una serie de tecnologías utilizadas para la realización de los experimentos y el análisis de los resultados. Seguidamente en el apartado 3.2 se describe el proceso seguido para generar los conjuntos de datos artificiales.
En el punto 3.3 se definen una serie de métricas que nos servirán para comparar los modelos, y las secciones 3.4, 3.5 y 3.6 presentan los resultados de los experimentos, para finalmente presentar una serie de conclusiones en el apartado 3.7.

3.1. Tecnologías utilizadas

Las tareas de programación necesarias para llevar a cabo los experimentos se han dividido en una serie de *scripts*, la mayoría de ellos escritos en *Python*, aunque también ha sido necesario implementar código en otros lenguajes como *Octave*, *AWK* o *BASH*.
Más en concreto, las tareas de generación de datos artificiales, cálculo de métricas, recolección de los resultados de los experimentos, entrenamiento de los modelos *ABR* y *SVR*, y visualización de los datos se han implementado en *Python*. Por contra, el entrenamiento de modelos *GPR* se ha realizado en *Octave*. El tratamiento de ficheros de texto se ha hecho mediante *AWK*, y mediante *bash scripting* se ha automatizado la ejecución de los experimentos.

3.2. Construcción del conjunto de datos artifical

Con el objetivo de obtener comparativas entre las predicciones de los diferentes modelos en un contexto donde la distribución del ruido es conocida, se obtienen muestras de un proceso aleatorio con varianza homocedástica (independiente de la entrada). El proceso se define de la siguiente manera:

$$y = f(x) + \varepsilon, \quad x \in \mathbb{R} \tag{3.1}$$

donde

$$f(x) = wx, \quad w \in \mathbb{R} \tag{3.2}$$

y

$$\varepsilon \sim \mathcal{N}\left(0, \sigma^2\right). \tag{3.3}$$

A continuación describimos el proceso mediante el cuál se han generado artificialmente las observaciones $\boldsymbol{y} = \{y_i\}_1^N$:

1. Se generan muestras aleatorias $x_i \sim \mathcal{U}\left(x_{min}, x_{max}\right)$.
   En concreto se elige $x_{min} = 0, x_{max} = 100$, pero obsérvese que la elección de la longitud y posición de este intervalo no es relevante en los experimentos, ya que al definir f como una función lineal, "moverse" y "acercarse" no modifican el problema.

2. Se elige aleatoriamente el coeficiente $w \sim \mathcal{U}\left(w_{min}, w_{max}\right)$.
   Los valores seleccionados fueron $w_{min} = 0, w_{max} = 1$.

3. De cara a mantener un nivel de ruido controlado se selecciona σ mediante la fórmula:

   $$\sigma = \left(\%ruido \cdot (x_{max} - x_{min}) \cdot \sqrt{2\pi} w\right)^{-1} = \frac{1}{5 \cdot \sqrt{2\pi} w} \simeq \frac{0{,}08}{w} \tag{3.4}$$

   La función de densidad de probabilidad de la variable aleatoria ε es

   $$f(x) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{3.5}$$

   que para $\sigma \neq 0$, alcanza en $x = 0$ su máximo: $\frac{1}{\sigma\sqrt{2\pi}}$.
   Por ello, eligiendo σ según la ecuación 3.4 con $\%ruido = 5\,\%$, el valor máximo para ε queda acotado por $\%ruido \cdot (x_{max} - x_{min}) \cdot w = 0{,}05 \cdot 100 \cdot w = 5w$.

4. Lo siguiente es generar el ruido, tomando las muestras aleatorias $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

5. Finalmente se obtienen las observaciones artificiales con ruido $y_i = wx + \varepsilon_i$.

Para los experimentos descritos en los siguientes apartados los valores obtenidos para w y σ fueron respectivamente 0,825 y 0,097, y se generaron un total de $5 \cdot 10^4$ muestras para entrenamiento y $5 \cdot 10^4$ muestras para test. El algoritmo de generación de estas observaciones artificiales se implementó en *Python*.

3.3. Calidad de las predicciones

Con el propósito de poder comparar los resultados obtenidos por los diferentes modelos es necesario definir una serie de funciones de error que nos indiquen con qué precisión se ajusta el modelo entrenado a los datos de test. En el problema propuesto, debemos diferenciar entre dos tipos de funciones de error o métricas: aquellas que miden el error en las predicciones y aquellas que miden el error en las estimaciones de los cuantiles.

3.3.1. Error en las predicciones de la variable de respuesta

Utilizaremos como métrica de la calidad en las predicciones el Error Cuadrático Medio (ECM) que es una de las más difundidas en el campo del aprendizaje automático.

Definición 8 *Sean* $\boldsymbol{y} = (y_1, ..., y_n)$ *un vector de valores reales,* $\hat{\boldsymbol{y}} = (\hat{y_1}, ..., \hat{y_n})$ *un vector de predicciones , se define el ECM entre ambos como*

$$ECM(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \tag{3.6}$$

En el caso de una predicción perfecta, este error es cero (el valor mínimo posible), mientras que por contra no hay una cota superior del valor que puede alcanzar este error.

3.3.2. Error en la estimación de los cuantiles

Puesto que no estamos únicamente interesados en las predicciones, sino también en dar bandas de confianza para las mismas, hemos de ser capaces de medir también la calidad de éstas. Con este fin, y dados $\boldsymbol{y} = (y_1, ..., y_n)$ un vector de valores reales, $\alpha \in [0, 1]$ y $\hat{\boldsymbol{q}} = (\hat{q_1}, ..., \hat{q_n})$ un vector de estimaciones del α-cuantil, se definen las siguientes métricas:

- **Ratio de Observaciones Fuera del Intervalo de Confianza(ROFIC)**: Se trata de una de las métricas más básicas que podemos utilizar. Consiste en un ratio calculado a partir del porcentaje de observaciones, o valores de respuesta reales, que quedan fuera del cuantil estimado.

  Definición 9 *(Ratio de Observaciones Fuera del Intervalo de Confianza)*

  Se define el error ROFIC como:

  $$ROFIC(\boldsymbol{y}, \hat{\boldsymbol{q}}, \alpha) = \left| \frac{\#\{i : y_i > \hat{q_i}\}}{n} - (1 - \alpha) \right| \tag{3.7}$$

Un modo simple de interpretar este error es el siguiente: la expresión de la izquierda en la resta, multiplicada por 100, representa el porcentaje de observaciones que no se encuentran dentro de las bandas de confianza estimadas. Y la expresión de la derecha, multiplicada por 100, es el porcentaje de observaciones que se espera estimar mal con una confianza del $100\alpha\,\%$. El valor óptimo de esta métrica es 0 y se alcanza cuando el porcentaje de puntos dentro de la banda de confianza es el esperado. Además, esta métrica está normalizada, de manera que está acotada superiormente por 1.

- **Error directo en la estimación del cuantil(EDEC)**:
  Si conocemos la distribución F_X de los errores de antemano podemos definir la siguiente función de error:

  Definición 10 *(Error directo en la estimación del cuantil)*
  Se define el error EDEC como:

$$EDEC(\boldsymbol{y}, \hat{\boldsymbol{q}}, \alpha) = \frac{\left|\left(\frac{1}{n}\sum_{i=1}^{n}(\hat{q}_i - y_i)\right) - Q_X(\alpha)\right|}{Q_X(\alpha)} \tag{3.8}$$

  donde Q_X es la función cuantil definida en 4.

Observar que, como en la métrica ROFIC, un modelo que estimase el cuantil de manera óptima obtendría un error de 0 con la métrica EDEC.

Como ya se ha descrito en la sección 3.2, en el conjunto de datos generado artificialmente los errores ε se distribuyen normalmente, por lo que el valor de los cuantiles reales puede ser aproximado, por ejemplo, en el lenguaje *Python* mediante el método *ppf* de la clase *norm* accesible desde el paquete *scipy.stats*.

- **Error de Pinball Medio(EPM)**:

  Definición 11 *(Función de Pinball)*
  Definimos la función de pérdida de Pinball como:

$$L_P(y, q, \alpha) = \begin{cases} \alpha(y-q) & : \quad y \geq q \\ (\alpha-1)(y-q) & : \quad y < q \end{cases} \tag{3.9}$$

  Se trata de una función que ya se ha utilizado extensamente en la literatura del aprendizaje automático para la estimación de cuantiles [3] [15] y la comparación de modelos [16].

Definición 12 *(Error de Pinball Medio)*
A partir de la función de Pinball definimos el EPM como

$$EPM(\boldsymbol{y}, \hat{\boldsymbol{q}}, \alpha) = \frac{1}{n} \sum_{i=1}^{n} L_P(y_i, \hat{q}_i, \alpha) \tag{3.10}$$

Al igual que en las métricas EDEC y ROFIC, el valor mínimo que se puede obtener para el valor del EPM es 0.

3.4. Resultados del modelo *GPR*

El funcionamiento del modelo *GPR* ha sido descrito en la sección 2.2. Puesto que se trata de una herramienta bastante extendida en los campos de estadística y aprendizaje automático, no ha sido necesaria su implementación, ya que existen una multitud de librerías de código libre que contienen los algoritmos necesarios para su aplicación. En [17] puede encontrarse una comparativa de las características que presentan tres de éstas librerías (*GPstuff, GPML y FBM*).

Para realizar los experimentos se consideró en primera instancia la utilización del módulo de procesos gaussianos que implementa la librería *scikit-learn* para el lenguaje de programación *Python*. No obstante, la elección final fue la implementación de código libre *GPstuff* [17], una versátil colección de herramientas computacionales para procesos gaussianos compatible con *Matlab* y *Octave*. El motivo de esta decisión se debe a la enorme flexibilidad y variedad que presenta la librería *Gpstuff* frente a sus competidores, ya que contiene una gran variedad de funciones de covarianza, la posibilidad de modelar el ruido con distribuciones no gaussianas y otra serie de características muy avanzadas.
Así pues, como primera toma de contacto con el código de *GPstuff*, se desarrolló un programa en *Octave* que realizara la inferencia sobre el conjunto de datos artificiales. Con este fin se eligió el modelo *GPR* más simple, es decir, con función de covarianza lineal (2.2.6) y se entrenó llevando a cabo una optimización de los hiperparámetros mediante la estimación del *MAP* maximizando la función de verosimilitud marginal tal y como se describe en (2.2.7), que en *Gpstuff* se realiza aplicando el método de descenso por gradiente *scaled conjugate gradient*.

En concreto se entrenaron un total de 43 modelos *GPR* cada uno con un número mayor de patrones de entrenamiento que el anterior (tamaños escogidos de acuerdo a una escala logarítmica en base e), hasta alcanzar el límite (debido a la capacidad de indexación) permitido por *Octave* de 6641

patrones de entrenamiento.
Para cada uno de estos modelos optimizados se realizaron predicciones con un conjunto de test formado por 50000 patrones.

A modo de comprobación de la calidad de las predicciones, se utilizó el modelo entrenado con el conjunto de entrenamiento más grande para calcular los errores (o residuos) de las predicciones. Puesto que los datos se generaron artificialmente, si el modelo ha sido capaz de inferir la función subyacente sabemos que estos residuos deben seguir una distribución normal. Una herramienta que nos permite comparar la distribución de estos residuos frente a una distribución teórica son las gráficas QQ, que en *Python* se consigue con el método *qqplot* de la API del módulo *statsmodel*. En la Figura 3.1 encontramos la gráfica QQ obtenida por *GPR*, que nos muestra que la distribución de los errores de las predicciones y la distribución son prácticamente idénticas.

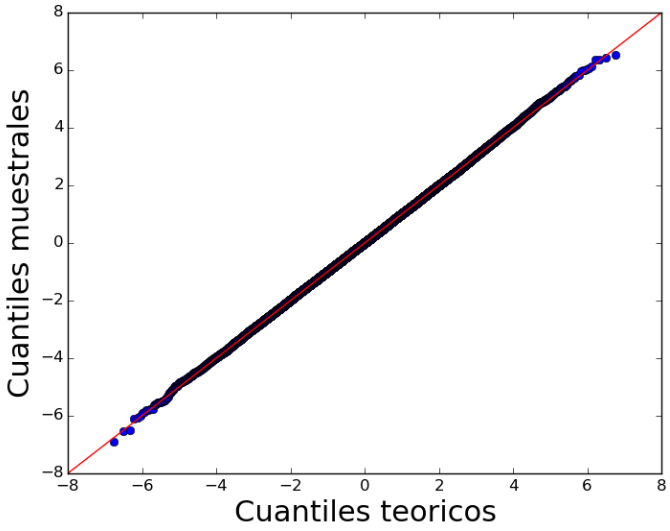


Figura 3.1: Gráfica Q-Q de los residuos de un modelo *GPR* entrenado con 6641 patrones de entrenamiento.

Además, a partir de los resultados obtenidos, se utilizaron una serie de *scripts* en *Python* para medir las métricas de error (definidas previamente en este capítulo) en cada uno de los modelos. Con estos valores se generaron una serie de gráficas que se presentan a continuación.

En la figura 3.2 encontramos la curva de aprendizaje del modelo *GPR*, es decir, la evolución del error ECM respecto al tamaño del conjunto de entrenamiento. Como puede observarse, el modelo alcanza rápidamente el mínimo

y se estabiliza para un número relativamente pequeño de patrones.

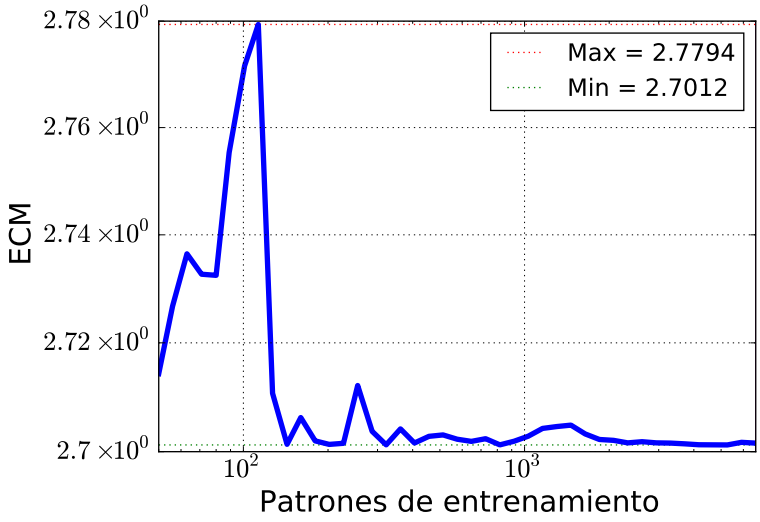
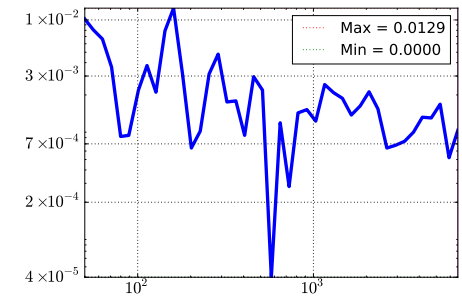
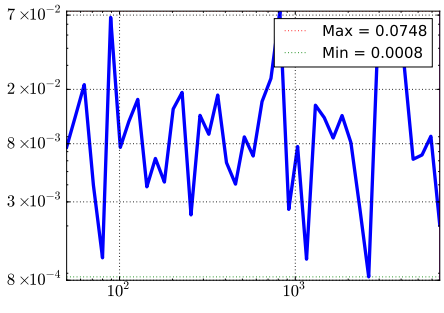


Figura 3.2: GPR - Curva de aprendizaje.

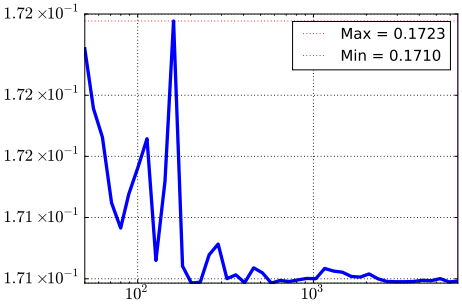
También se han obtenido los valores (para distintos tamaños del conjunto de entrenamiento) de las métricas de error para el cuantil 95 % (ejemplos de otros cuantiles pueden encontrarse en el Anexo A) tal y como se muestra en la Figura 3.3.



(a) ROFIC



(b) EDEC



(c) EPM

Figura 3.3: GPR - Medidas de error frente al tamaño del conjunto de entrenamiento en la estimación del 95-cuantil.

Notar que el comportamiento de la métrica EPM es parecido al previamente visto en la curva de aprendizaje, el modelo rápidamente se estabiliza. Sin embargo, las otras dos métricas no parecen presentar ninguna tendencia.

3.5. Resultados del modelo *ABR*

Para realizar los experimentos la implementación de *ABR* elegida ha sido la que proporciona la librería *scikit-learn* escrita en *Python*. Este modelo presenta una serie de parámetros (número de árboles, tasa de aprendizaje, etc.) que deben fijarse, además de los parámetros de los árboles de regresión (profundidad máxima, número mínimo de patrones por hoja, etc.). Para ajustar los mismos, se hizo un muestreo aleatorio de 100 combinaciones dentro de un rango de valores considerados y se eligió aquel modelo que presentaba mejor rendimiento en un conjunto de validación.
Se realizaron las mismas pruebas que las llevadas a cabo con *GPR* descritas

en (3.4), con los mismos conjuntos de entrenamiento y test y los resultados obtenidos se presentan a continuación.
En la Figura3.4 encontramos la gráfica QQ de los residuos de las predicciones de este modelo, que no se ajustan tan bien a los valores teóricos como en el caso del modelo *GPR*.

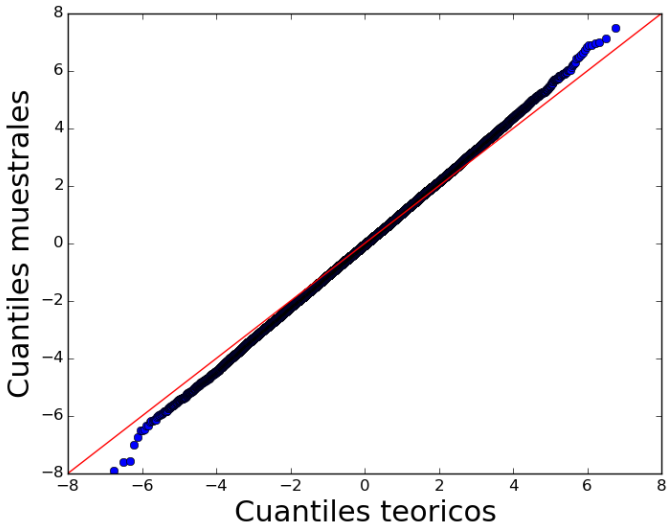


Figura 3.4: Gráfica Q-Q de los residuos de un modelo *ABR* entrenado con 6641 patrones de entrenamiento.

Cabe destacar que este modelo, al estar implementado en *Python*, no sufre de las limitaciones de memoria presentes en *Octave*, por lo que se pudo entrenar modelos con conjuntos de entrenamiento más grandes (hasta un máximo de 15000 patrones de entrenamiento). Esto se refleja en las gráficas que se muestran a continuación donde podemos observar una linea morada vertical que marca el límite del tamaño del conjunto de entrenamiento (6641 patrones) alcanzado por *GPR*.

En la Figura 3.5 observamos que el modelo *ABR*, para obtener resultados de predicción similares a los obtenidos por *GPR* (ECM $\simeq$ 2,8), necesita conjuntos de entrenamiento mucho más grandes, lo cual puede ser un inconveniente muy importante al tratar con datos extraídos de problemas reales.

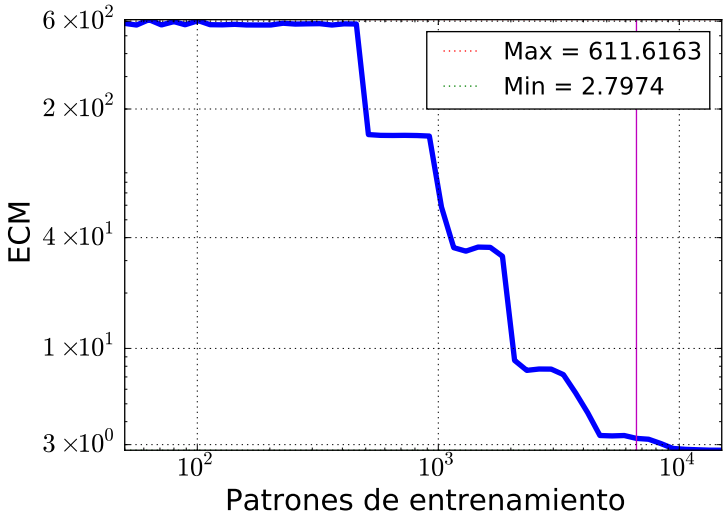
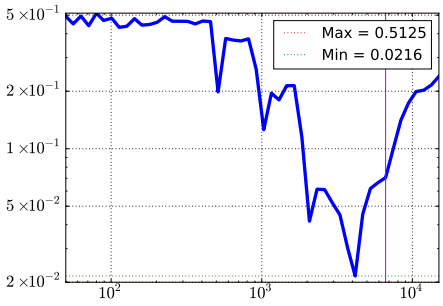
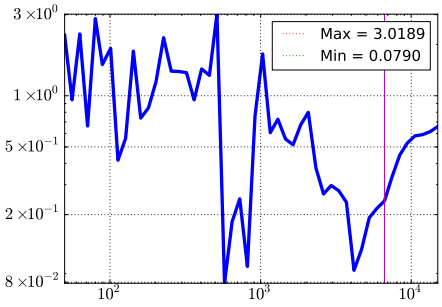


Figura 3.5: ABR - Curva de aprendizaje.

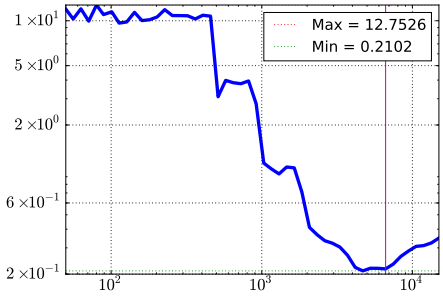
En la Figura 3.6 se muestran los valores de las métricas utilizadas para medir la calidad de la estimación del cuantil 95 % frente al tamaño del conjunto de entrenamiento. Se observa un dato curioso: antes de llegar a la línea vertical que representa un conjunto de entrenamiento de 6641 patrones, el modelo empieza a sobreajustar las estimaciones de los cuantiles, ya que para los cuantiles seleccionados, las tres métricas alcanzan su mínimo y después aumentan en el mismo punto.
Para intentar paliar este comportamiento sería necesario un esfuerzo adicional para imponer una mayor regularización sobre los árboles, o aplicar otra medida para controlar el sobreajuste. Sin embargo no nos encargaremos en este trabajo de realizar esta labor, ya que queda fuera de los límites del mismo. En cualquier caso es destacable que a pesar de que para el modelo GPR no se ha realizado ningún esfuerzo en esta línea, no se han observado problemas de sobreajuste.



(a) ROFIC



(b) EDEC



(c) EPM

Figura 3.6: ABR - Medidas de error frente al tamaño del conjunto de entrenamiento en la estimación del 95-cuantil.

3.6. Resultados del modelo *SVR*

Para este modelo se ha elegido al igual que con el modelo *ABR*, la implementación de la librería *scikit-learn*. En concreto, se trata del modelo ϵ-*Support Vector Regression* que está basado en el modelo *LIBSVM*. Los parámetros libres que deben ajustarse son C y ϵ, que controlan el sobreajuste del modelo. Al igual que para el modelo *ABR*, para ajustar los parámetros se hizo un muestreo aleatorio de 100 combinaciones y se eligió el mejor modelo mediante un conjunto de validación.
La gráfica QQ obtenida con este modelo se muestra en la Figura 3.7 y como podemos observar tampoco llega a ajustarse tan bien como la del modelo *GPR*.

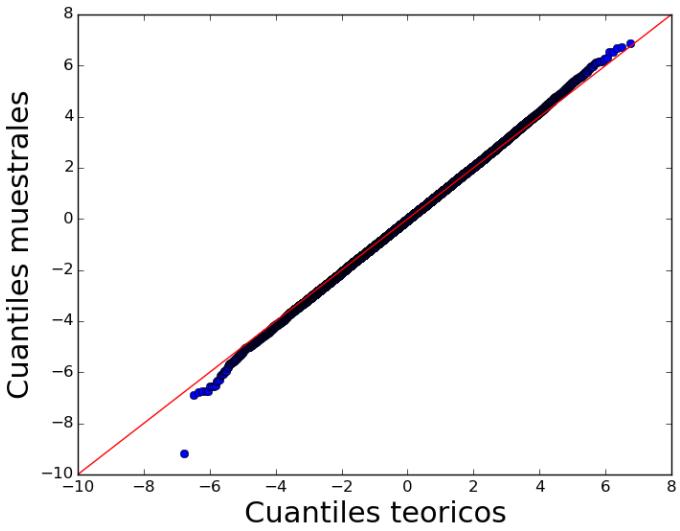


Figura 3.7: Gráfica Q-Q de los residuos de un modelo *SVR* entrenado con 6641 patrones de entrenamiento.

Del mismo modo que para los modelos anteriores, se presenta en la Figura 3.8 la curva de aprendizaje del modelo *SVR*. En esta ocasión observamos que para conjuntos de entrenamiento pequeños el modelo mejora lentamente. Llegado un punto, el modelo mejora de manera muy rápida hasta que a partir de los 1000 patrones de entrenamiento aproximadamente, el modelo converge a su capacidad máxima de generalización.

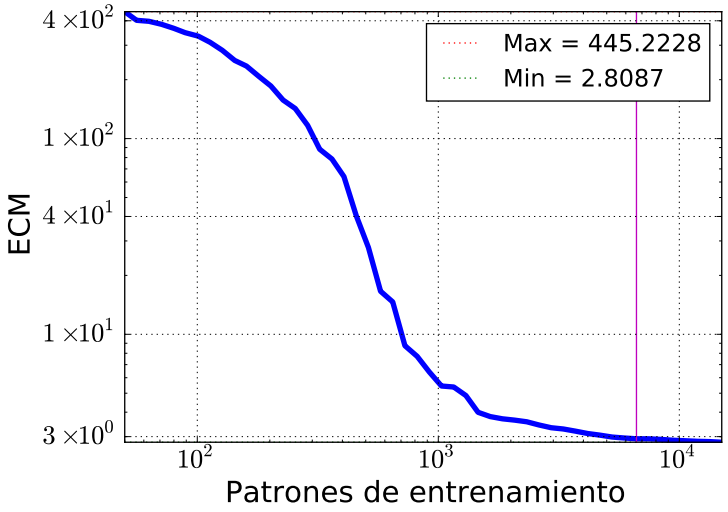
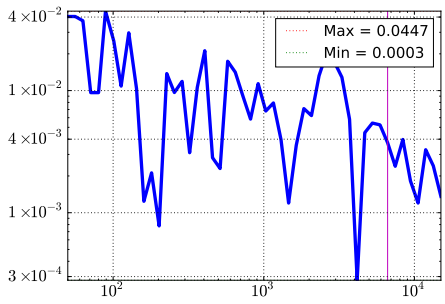
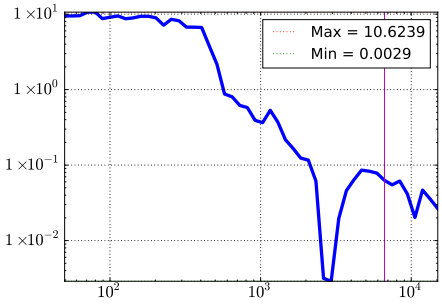


Figura 3.8: SVR - Curva de aprendizaje.

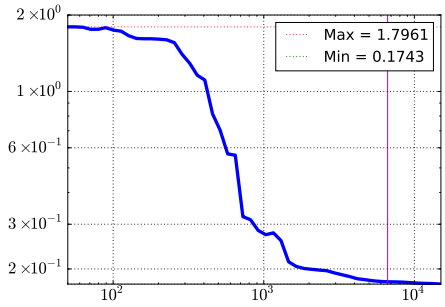
También se presenta, como se hizo para $GPR(3.4)$ y $ABR(3.5)$, en la Figura 3.9 la evolución de las métricas para la calidad de la estimación del cuantil 95 % con respecto al tamaño del conjunto de entrenamiento.



(a) ROFIC



(b) EDEC



(c) EPM

Figura 3.9: SVR - Medidas de error frente al tamaño del conjunto de entrenamiento en la estimación del 95-cuantil.

Observamos que las tres métricas muestran una tendencia descendente, es decir, el modelo gana capacidad de predicción conforme más datos de entrenamiento se le proporcionan. No obstante, esta mejora es lenta.

3.7. Conclusiones extraídas de los experimentos

A la vista de los resultados, se puede determinar que los tres modelos presentan un rendimiento muy parecido en lo que respecta a la predicción de la respuesta, aunque el de *GPR* es algo mejor (ECM $\simeq 2{,}7$ para *GPR* y $\simeq 2{,}8$ para los otros dos). No obstante, cabe destacar que mientras que *ABR* y *SVR* tienen una curva de aprendizaje que mejora lentamente, *GPR* es capaz de obtener mejores resultados empleando un número mucho menor de patrones de entrenamiento.

En cuanto a la estimación de cuantiles, el mejor rendimiento de los tres mo-

delos lo proporciona *GPR*, seguido de cerca por *SVR*, mientras que *ABR* encuentra mayores dificultades al estimar la incertidumbre en sus predicciones, probablemente debido a que infravalora las estimaciones de los cuantiles, situación que las métricas ROFIC y EPM penalizan mucho más que si se sobrevaloran.

Con estas consideraciones, parece que el modelo *GPR* es extremadamente eficaz prediciendo la distribución de la variable de respuesta. Esto puede deberse a que los datos artificiales cumplen las hipótesis (descritas en 2.2.4) en las que se basa este modelo y a la elección adecuada de la función de covarianza (lineal) para modelar la variable de respuesta (que es una función lineal de las entradas). Debido a sus características particulares, *GPR* es capaz de aprovechar esta información a priori sobre los datos, cosa que no pueden hacer *ABR* ni *SVR*. Con esto *GPR* parece demostrar las propiedades teóricas que posee: proporcionando información a priori, el modelo tiene mucho potencial.
No obstante, lo más normal es no tener ninguna (o muy poca) información a priori sobre los datos, por lo que no se puede determinar qué modelo es más adecuado para la resolución del problema de predicción de la distribución de la demanda hasta realizar experimentos con datos reales.

4 Experimentos con datos de sucursales bancarias

En este capítulo se presentan los resultados obtenidos por los modelos de regresión en los experimentos con datos de un problema real.
En primer lugar, se describe en la sección 4.1 la estructura de los conjuntos de datos utilizados en los experimentos. Seguidamente, en el punto 4.2 se enumeran las distintas métricas que nos servirán para comparar los modelos, y en las secciones 4.3, 4.4 y 4.5 se explica el proceso seguido para entrenar los modelos y se muestran algunas gráficas de ejemplo. Finalmente, los resultados obtenidos se comparan en el apartado 4.6.

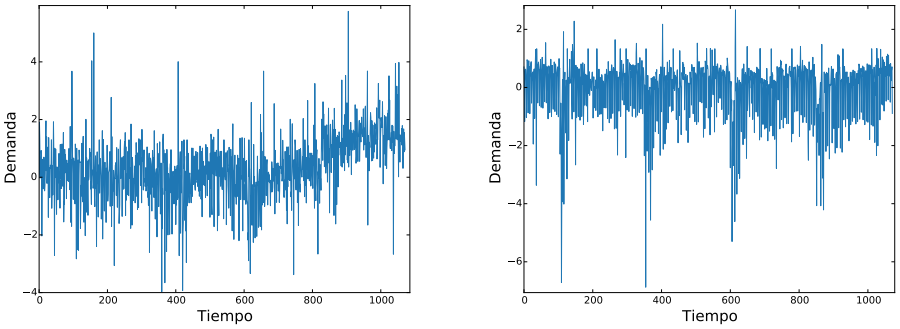
4.1. Descripción de los conjuntos de datos

Los datos utilizados para realizar los experimentos descritos en este capítulo son datos recogidos de sucursales bancarias reales para las que se quiere predecir la distribución de la demanda diaria de dinero en efectivo por parte de sus clientes.

En concreto, se consideran un total de 46 sucursales para cada una de las cuáles se tiene un conjunto de datos diferente.
En cada uno se recogen, aproximadamente, un total de 1073 patrones (excepto dos sucursales que presentan 1048) formados por 186 variables predictoras (características) y la demanda (variable de respuesta).
Los conjuntos con 1073 patrones se han dividido en un conjunto de entrenamiento con 750 patrones ($\sim 70\,\%$) y otro de test con 322 patrones ($\sim 30\,\%$).
En la división de los conjuntos con 1048 patrones se ha mantenido esta proporción, obteniendo un conjunto de entrenamiento con 733 patrones y otro de test con 315 patrones.
Esta división se ha hecho preservando el orden temporal de los patrones, ya que al tratarse de un problema de series temporales, es importante entrenar con los patrones más antiguos y realizar las pruebas con los más recientes.
A cada par entrenamiento-test se le ha aplicado un proceso de normalización de las características de modo que tengan media 0 y desviación típica

1. Además, la variable de respuesta ha sido estandarizada de modo que se encuentre acotada entre 0 y 1.

En la Figura 4.1 pueden verse ejemplos de series temporales de las demandas a lo largo del tiempo en dos de estas sucursales, una muy difícil de predecir (sucursal 34) y otra relativamente fácil de predecir (sucursal 44). En las diferentes secciones de este capítulo se muestran gráficas de los resultados obtenidos por los modelos en estas mismas sucursales.



(a) Sucursal 34 (b) Sucursal 44

Figura 4.1: Demanda respecto al tiempo en sucursales reales.

4.2. Calidad de las predicciones

En la sección 3.3.1 se presentó la métrica ECM para el error de las predicciones. Sin embargo, en los experimentos con datos de sucursales, utilizaremos una función de error distinta para medir la calidad de las predicciones de la variable de respuesta. Se trata de la función de Error Absoluto Medio (EAM).

Definición 13 *(Error Absoluto Medio)*
Sean $\boldsymbol{y} = (y_1, ..., y_n)$ *un vector de valores reales,* $\hat{\boldsymbol{y}} = (\hat{y_1}, ..., \hat{y_n})$ *un vector de predicciones , se define el EAM entre ambos como*

$$EAM(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y_i}| \tag{4.1}$$

En el caso de una predicción perfecta, este error es cero (el valor mínimo posible), mientras que por contra no hay una cota superior del valor que puede alcanzar este error.

Este cambio en la métrica utilizada se debe a que hemos considerado que una función de error lineal representa mejor que una función cuadrática la situación que se da en las sucursales bancarias, dado que la magnitud en los errores de planificación se relaciona linealmente con los costes de operación que suponen.

Además de esta función de error, en los experimentos descritos en este capítulo se ha tenido en cuenta otro baremo para comparar los modelos. Se trata del coeficiente R^2, también conocido como coeficiente de determinación.

Definición 14 *(Coeficiente de determinación)*
Dado el conjunto de valores objetivo $y_1, ..., y_n$ y un conjunto asociado de predicciones $\hat{y}_1, ..., \hat{y}_n$, y denotando como $\bar{y}$ a la media empírica de los valores y_i, el coeficiente R^2 se define del siguiente modo:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{4.2}$$

donde

$$SS_{tot} = \sum_i (y_i - \bar{y})^2, \tag{4.3}$$

$$SS_{res} = \sum_i (y_i - \hat{y}_i)^2. \tag{4.4}$$

El objetivo de este coeficiente es medir la capacidad de generalización del modelo, de manera que el máximo (y mejor) valor posible es 1. Una característica importante es que este coeficiente puede ser negativo, lo que indicaría que el modelo tiene menor capacidad predictora que un modelo que estime siempre mediante la media empírica.

Respecto a la estimación de los cuantiles, se han medido las métricas ROFIC y EPM definidas en 3.3.2. La métrica EDEC, sin embargo, no es aplicable a estos experimentos ya que la distribución real de las demandas es desconocida.
Todas las métricas mencionadas en esta sección han sido calculadas para las predicciones obtenidas mediante una serie de *scripts* en *Python*.

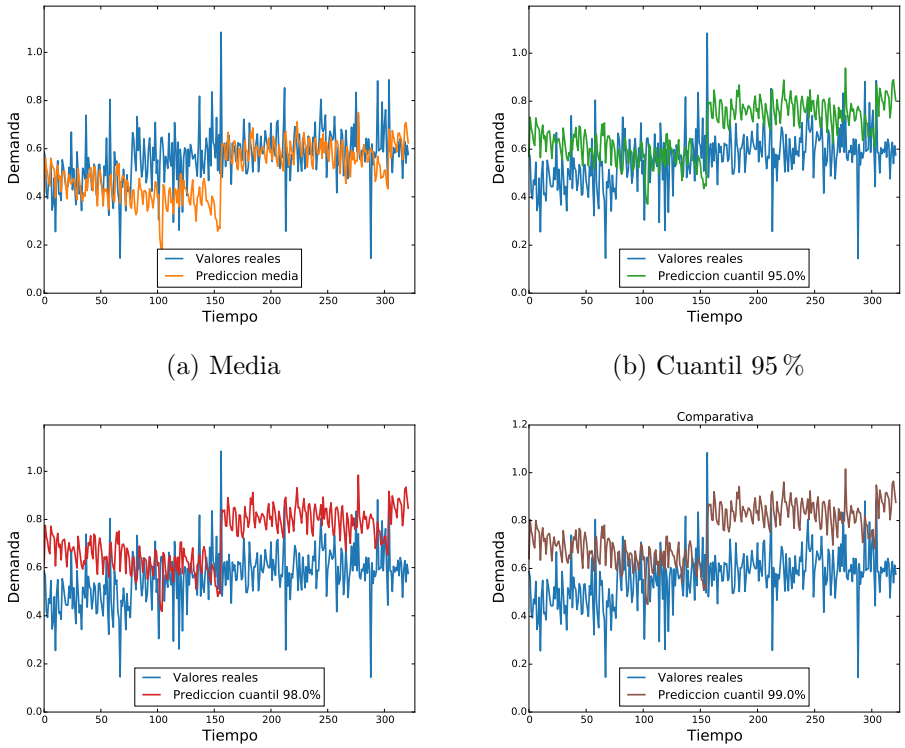
4.3. Experimentos con *GPR*

Al igual que con los experimentos sobre datos artificiales, se utilizó la librería *GPstuff* para realizar las pruebas con datos de sucursales. Esta vez,

se eligió como función de covarianza la función cuadrática exponencial, que es la elección más común cuando no se tiene información a priori sobre las propiedades de la función subyacente. Además, el modelo *SVR* utiliza este mismo *kernel* (bajo el nombre de *kernel RBF*) por lo que se considera que así los modelos están en mayor igualdad de condiciones frente al problema. Se ha elegido la forma anisotrópica de ésta función, ya que proporciona mayor potencia al modelo.

Durante las pruebas se vió que al realizar el ajuste de los hiperparámetros mediante la maximización de la log-verosimilitud, la calidad de las predicciones dependía en gran medida del valor inicial elegido para comenzar el descenso por gradiente. Una posible explicación a este fenómeno es que el problema de ajuste de los hiperparámetros está plagado de mínimos locales, por lo que empezar el descenso en distintos puntos lleva a mínimos locales distintos que no tienen porque dar soluciones buenas.
Por esta razón, a la hora de realizar la búsqueda de parámetros, se entrenaron para cada sucursal 20 modelos *GPR* distintos, de manera que los valores iniciales de los hiperparámetros se escogieron aleatoriamente de un conjunto de valores considerados. Una vez realizada la optimización mediante el método de descenso por gradiente, se sacaron las predicciones de los 20 modelos para los conjuntos de test y se seleccionó para la comparativa con *SVR* y *ABR* aquel que presentó un EAM más bajo.

En las Figuras 4.2 y 4.3 se muestran las predicciones y estimaciones de algunos cuantiles del modelo *GPR* para las sucursales tomadas como ejemplo en la Figura 4.1.



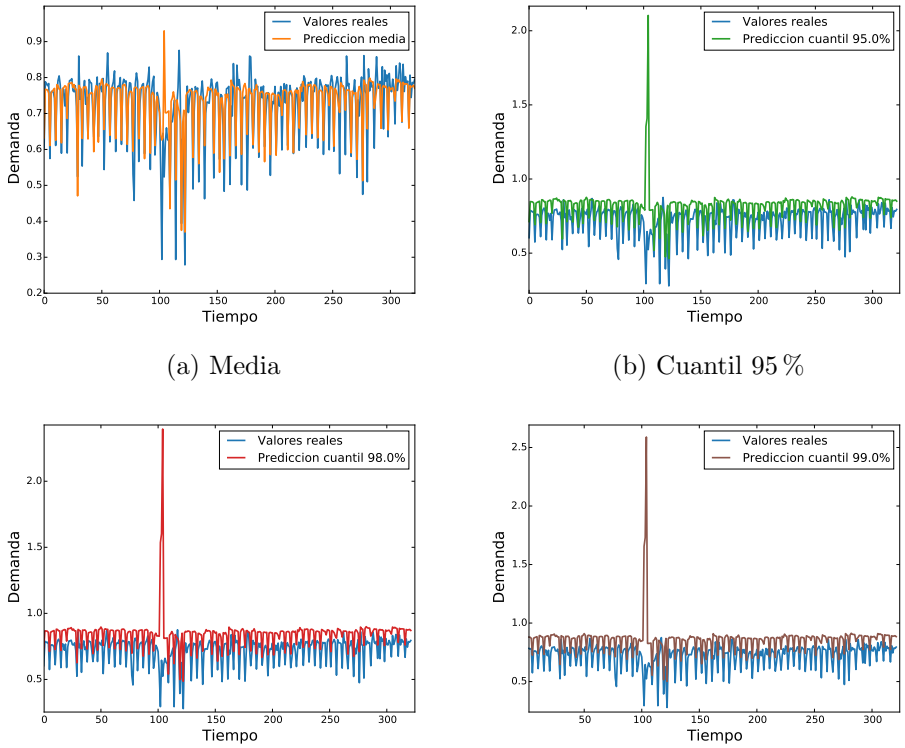
(a) Media

(b) Cuantil 95 %

(c) Cuantil 98 %

(d) Cuantil 99 %

Figura 4.2: GPR - Predicción frente a demanda en la sucursal 34.



(a) Media

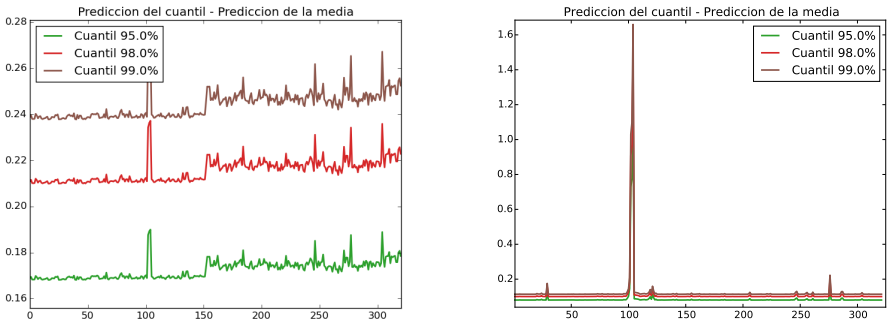
(b) Cuantil 95 %

(c) Cuantil 98 %

(d) Cuantil 99 %

Figura 4.3: GPR - Predicción de la demanda en la sucursal 44.

A simple vista, en las gráficas de las predicciones de la sucursal 34 puede parecer que el modelo está considerando la incertidumbre en las predicciones constante en el tiempo, pues parece que las estimaciones de los cuantiles son las predicciones desplazadas verticalmente. Sin embargo, en las gráficas de la sucursal 44 vemos claramente un pico muy pronunciado en la estimación de los cuantiles. Para ver qué está ocurriendo realmente, en la Figura 4.4 se muestra la incertidumbre (el valor que queda al restarle la predicción de la demanda a la estimación del cuantil) predicha por el modelo para las predicciones de las Figuras 4.2 y 4.3.



(a) Sucursal 34 (b) Sucursal 44

Figura 4.4: GPR - Incertidumbre en las predicciones de la demanda en sucursales reales.

Queda claro que la incertidumbre no es constante, ya que presenta picos y oscilaciones. Sin embargo, el rango en el que se mueven estas incertidumbres para las predicciones de la sucursal 34 es muy pequeño en comparación al valor de las predicciones, lo que provoca que parezcan constantes a simple vista.
Además, se observa que en la sucursal difícil (34) la incertidumbre es mayor y presenta más oscilaciones conforme evaluamos patrones más recientes en el tiempo. Una interpretación para esto es que la distribución de la demanda cambia conforme avanzamos en el tiempo, y puesto que el modelo ha sido entrenado con los patrones más antiguos, va perdiendo capacidad de predicción.

4.4. Experimentos con *ABR*

Todos los métodos que se describen en esta sección se han implementado en el lenguaje *Python*, las gráficas se han generado mediante *scripts* en este lenguaje y nuevamente se ha utilizado la implementación de este modelo presente en la librería para *scikit-learn*, que ya se utilizó para realizar los experimentos con datos artificiales.
Tal y como sucedía en los experimentos previos, debemos elegir un conjunto de valores para los parámetros libres del modelo. Sin embargo, la forma de elegirlos es un tanto diferente. Para los experimentos con datos de sucursales, se muestrean aleatoriamente 1000 combinaciones de los parámetros considerados y para cada combinación se aplica una técnica de validación cruzada. Las técnicas clásicas de validación cruzada (*LPO*, *LOO*, *k-fold*) no parecen adecuadas para aplicarlas a este problema, ya que no tienen en cuenta la importancia del orden de los patrones debido al carácter temporal

del problema. Por ello se diseña una técnica de validación cruzada de *ventana.*

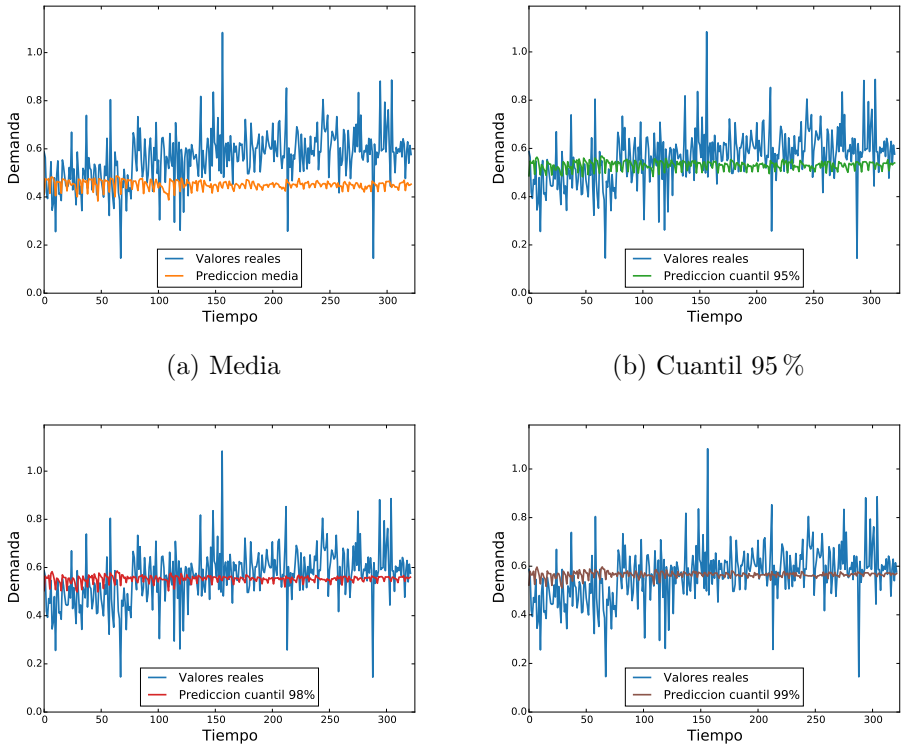
Para ello se definen los valores $m = 0{,}5 \cdot n$ y $d = 0{,}2 \cdot n$, donde n es el número de patrones en el conjunto de entrenamiento. Así, el método consiste en entrenar el modelo con los primeros m patrones de entrenamiento y medir su rendimiento en la predicción de los d siguientes.

A continuación, se re-entrena el modelo desplazando la ventana, es decir, descartando los d patrones iniciales, del conjunto de $n - d$ patrones que nos quedan tomamos los m primeros para entrenar y los d siguientes para evaluar. Se sigue este proceso repetidamente hasta llegar a evaluar los patrones del final del conjunto.

En total, se entrena y evalúa el modelo 3 veces, y se le asigna como rendimiento final la media del rendimiento obtenido en cada paso de la validación.

Finalmente, de las 1000 combinaciones muestreadas, se elige aquella que proporciona un mayor rendimiento en la validación cruzada para llevar a cabo las pruebas sobre el conjunto de test.

En las Figuras 4.5 y 4.6 se muestran las predicciones y estimaciones del modelo *ABR* para las sucursales 34 y 44.



(a) Media

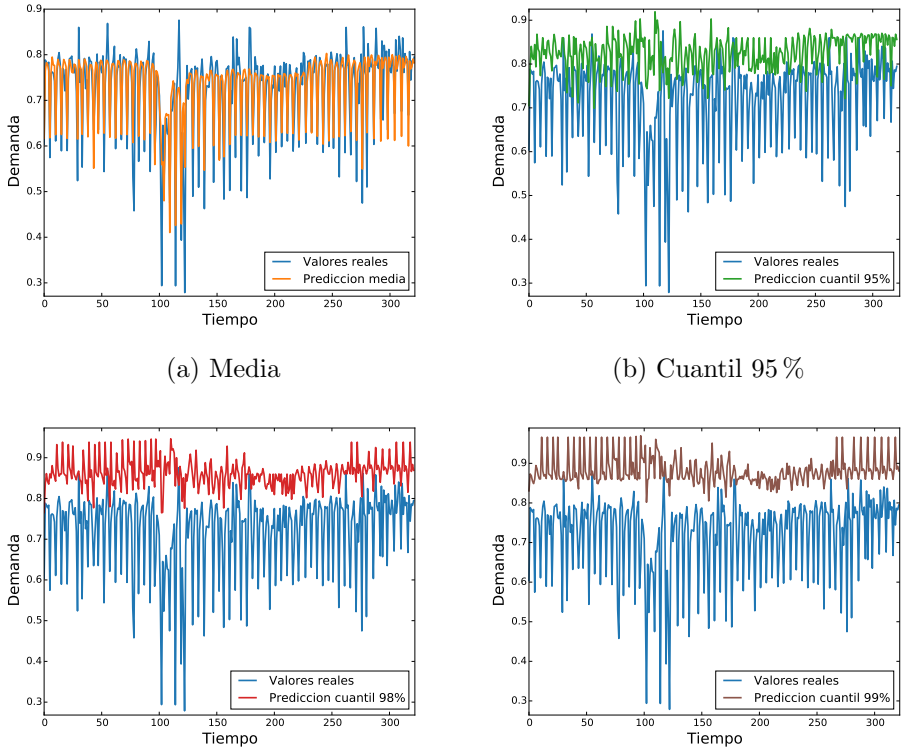
(b) Cuantil 95 %

(c) Cuantil 98 %

(d) Cuantil 99 %

Figura 4.5: ABR - Predicción frente a demanda en la sucursal 34.

En comparación a los resultados obtenidos con *GPR* para la sucursal 34, vemos que el modelo *ABR* no es capaz de predecir la demanda con tanta exactitud y que además tiende a infravalorar las estimaciones de los cuantiles, resultado que ya se preveía en las conclusiones de los experimentos con datos artificiales (3.7) y que, presumiblemente, hará que las métricas ROFIC y EPM le atribuyan un rendimiento pobre.



(a) Media

(b) Cuantil 95 %

(c) Cuantil 98 %

(d) Cuantil 99 %

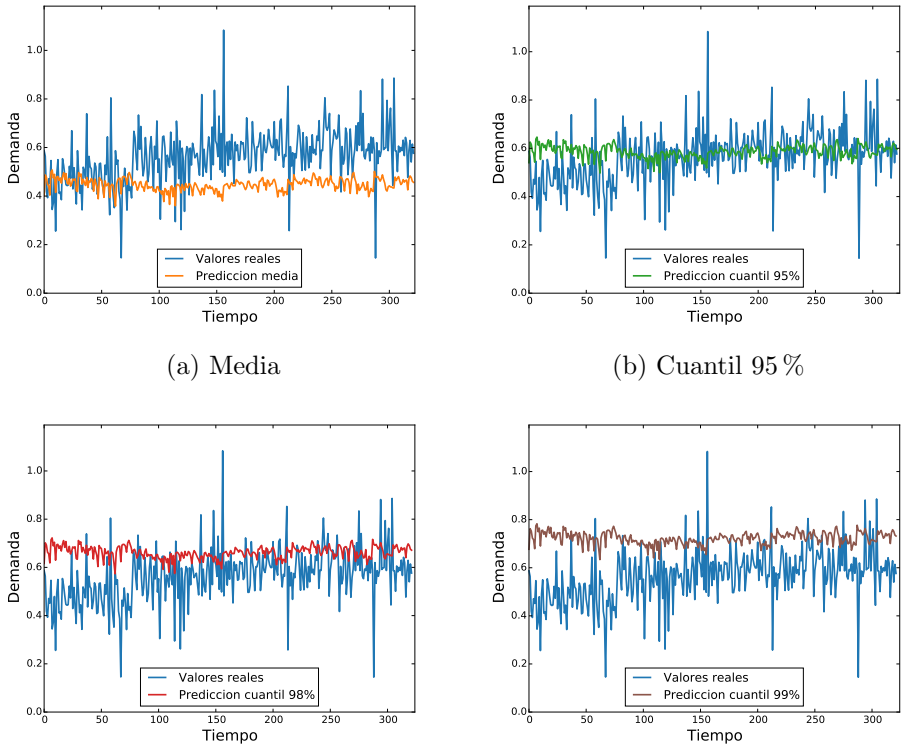
Figura 4.6: ABR - Predicción de la demanda en la sucursal 44.

Respecto a la sucursal 44, el modelo parece predecir bastante bien la tendencia del mismo, pero no se puede determinar a simple vista si las predicciones obtenidas para la demanda y los cuantiles de la distribución son mejores o peores que los obtenidos por *GPR*.

4.5. Experimentos con *SVR*

Para realizar los experimentos sobre datos de sucursales con el modelo *SVR* se ha utilizado la misma implementación que para las pruebas con datos artificiales descritas en la sección 3.6. De nuevo, las gráficas se han generado mediante *scripts* en *Python*.
El método de elección de los valores para los parámetros libres del modelo ha sido el mismo que el utilizado para *ABR* descrito en el apartado 4.4.

Los gráficas de predicción obtenidas por *SVR* para las sucursales tomadas en este capítulo como ejemplo se muestran en las Figuras 4.7 y 4.8.



(a) Media

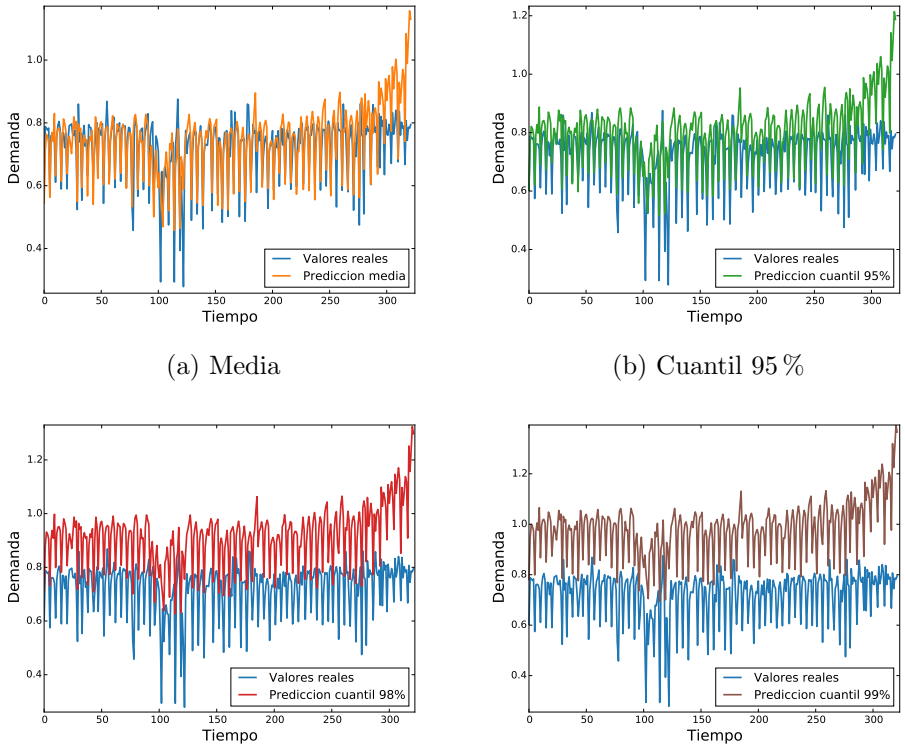
(b) Cuantil 95 %

(c) Cuantil 98 %

(d) Cuantil 99 %

Figura 4.7: SVR - Predicción frente a demanda en la sucursal 34.

En la Figura 4.7 vemos que el modelo tiene poca capacidad para predecir la demanda en la sucursal 34, pero que las estimaciones de los cuantiles parecen más adecuadas que las obtenidas con *ABR*, aunque no podemos determinar si son mejores o peores que las estimaciones de *GPR* hasta que comparemos las métricas consideradas.



(a) Media

(b) Cuantil 95 %

(c) Cuantil 98 %

(d) Cuantil 99 %

Figura 4.8: SVR - Predicción de la demanda en la sucursal 44.

Por contra, en la sucursal 44 el modelo parece predecir la demanda con bastante precisión pero se observa claramente que va perdiendo capacidad de generalización conforme avanzamos en el tiempo. Esto se debe, probablemente, a la característica ya comentada de que la distribución de la demanda cambia con el tiempo.

4.6. Comparativa de los resultados obtenidos

El objetivo de esta sección es presentar una comparativa de los resultados obtenidos por los tres modelos (*GPR*, *ABR* y *SVR*) en la predicción y estimación de incertidumbres de la demanda. En primer lugar, comparamos los modelos fijándonos en las sucursales de ejemplo 34 y 44 y seguidamente llevamos a cabo la comparación teniendo en cuenta el total de las 46 sucursales bancarias estudiadas.

De nuevo, tomando como ejemplo las sucursales 34 y 44, se presentan los Cuadros 4.1, 4.2 y 4.3, que nos muestran los valores de las métricas de

rendimiento obtenidas por cada modelo.

ID	R^2			EAM		
-	ABR	SVR	GPR	ABR	SVR	GPR
34	-1.00393	-1.12980	-0.58735	0.13165	0.13596	0.10043
44	0.74729	0.38670	0.65203	0.03235	0.05332	0.03722

Cuadro 4.1: R^2 y EAM obtenidos por cada modelo en las sucursales 34 y 44.

Tal y como se conjeturaba en las secciones 4.3, 4.4 y 4.5 las métricas consideradas confirman que en la sucursal 34 el modelo con mayor capacidad de predicción de la demanda es *GPR*, destacando notablemente frente a sus competidores.
Respecto a la sucursal 44, no se podía determinar a partir de las gráficas de las predicciones qué modelo daba mejores resultados en la predicción de la demanda. Los datos mostrados en 4.1 determinan que el modelo con mayor capacidad para predecir la demanda es *ABR*, seguido por *GPR* y obteniendo *SVR* el rendimiento más bajo.

ID	cuantil 95 %			cuantil 98 %			cuantil 99 %		
-	ABR	SVR	GPR	ABR	SVR	GPR	ABR	SVR	GPR
34	0.05885	0.03286	0.02165	0.05982	0.01291	0.01209	0.06014	0.00658	0.00802
44	0.00577	0.00677	0.00558	0.00263	0.00389	0.00269	0.00158	0.00262	0.00155

Cuadro 4.2: EPM obtenido por cada modelo para los cuantiles 95 %, 98 % y 99 % en las sucursales 34 y 44.

ID	cuantil 95 %			cuantil 98 %			cuantil 99 %		
-	ABR	SVR	GPR	ABR	SVR	GPR	ABR	SVR	GPR
34	0.59596	0.38789	0.14565	0.62596	0.12596	0.09491	0.63596	0.05211	0.06453
44	0.02205	0.03696	0.02826	0.00795	0.02000	0.01379	0.01795	0.01000	0.00689

Cuadro 4.3: ROFIC obtenido por cada modelo para los cuantiles 95 %, 98 % y 99 % en las sucursales 34 y 44.

Los resultados mostrados en los Cuadros 4.2 y 4.3 nos permiten afirmar que el rendimiento de *ABR* en la estimación de cuantiles para la distribución

de la demanda en la sucursal 34 es muy bajo en comparación al obtenido por *GPR* y *SVR*. De estos dos, *GPR* obtiene el mayor rendimiento para la estimación de los cuantiles 95 % y 98 % mientras que *SVR* le supera en el cuantil 99 %.
Respecto a la sucursal 44, *ABR* obtiene el mejor rendimiento en la estimación del cuantil 98 %, *GPR* en la estimación del cuantil 99 %, y para el cuantil 95 %, EPM y ROFIC apuntan a modelos distintos como el candidato al mayor rendimiento.

Estas consideraciones parecen determinar que *GPR* presenta algunas ventajas frente a sus competidores, pero no podemos basarnos únicamente en los resultados para dos casos de prueba para obtener una conclusión. Debemos hacer una comparativa entre las predicciones de todas las sucursales.
Con este fin, en el Cuadro 4.4 se presenta, para cada métrica medida, el promedio obtenido por las predicciones de cada modelo (en el Anexo B pueden encontrarse los valores obtenidos por los modelos en cada sucursal).

Métrica	ABR	SVR	GPR
R^2	0.06653	-0.01722	0.07097
EAM	0.05924	0.06273	0.05877
EPM (cuantil 95 %)	0.01407	0.01058	0.00958
EPM (cuantil 98 %)	0.01161	0.00532	0.00514
EPM (cuantil 99 %)	0.01079	0.00334	0.00325
ROFIC (cuantil 95 %)	0.13829	0.05727	0.02611
ROFIC (cuantil 95 %)	0.16383	0.02189	0.01649
ROFIC (cuantil 95 %)	0.17373	0.01588	0.01333

Cuadro 4.4: Promedio de los valores de las métricas obtenidos por los modelos de predicción.

Observamos que *GPR* supera (en media) a *ABR* y *SVR* en todas las métricas consideradas, por lo que ahora sí podemos determinar que se trata del modelo que proporciona un mayor rendimiento a la hora de extraer información sobre la distribución de la demanda en las sucursales estudiadas. Cabe decir que, aunque *GPR* obtiene un rendimiento superior que el alcanzado por los otros dos modelos, también *GPR* es el modelo que necesita tiempos de entrenamiento más altos, ya que su complejidad básica es de $O(n^3)$ siendo n el número de patrones de entrenamiento. Para el caso experimental estudiado este problema no es tan grave, ya que los conjuntos de datos son relativamente pequeños.

5 Conclusiones y trabajo futuro

Este trabajo ha intentado arrojar algo de luz a la hora de decidir qué modelo de regresión nos puede proporcionar una mejor previsión de las demandas futuras y su distribución dentro del problema del control del inventario. Para ello, se realizaron pruebas con los modelos *GPR*, *ABR* y *SVR* (que aportan enfoques distintos de cómo obtener estas predicciones) utilizando datos artificiales y datos de sucursales reales. Finalmente, los resultados obtenidos se compararon de acuerdo a una serie de métricas definidas sobre las predicciones.

De los resultados obtenidos con los experimentos sobre datos reales se extrajo la conclusión de que el modelo *GPR* es muy potente cuando se cumplen las hipótesis que este asume. Decimos que es un modelo potente porque además de obtener los mejores resultados, se necesitó un número de patrones de entrenamiento muy pequeño (en comparación a los necesitados por *ABR* y *SVR*) para obtener buenos resultados en las predicciones.

Tras los experimentos con datos reales de sucursales bancarias, se observó que nuevamente el modelo *GPR* presentaba (en media) los mejores resultados de los tres modelos estudiados.
En el análisis de estos resultados notamos también que el modelo *ABR* obtenía rendimientos mucho peores a la hora de estimar los cuantiles de la distribución de la demanda, a pesar de que el rendimiento en la predicción de ésta es mejor que el de *SVR*.
Podemos explicar este fenómeno fijándonos en que las medidas de error que se consideraron (EPM y ROFIC) perjudican, en el sentido de que asignan un rendimiento peor, a aquellos modelos cuyas estimaciones de los cuantiles quede por debajo del cuantil real (caso del modelo *ABR*, que tiende a infravalorar estas estimaciones).
En el caso particular estudiado de predicción de la demanda en sucursales bancarias, tiene sentido asumir este planteamiento, ya que es peor no disponer de efectivo suficiente para satisfacer la demanda que tener excedentes. Sin embargo, esto no tiene por qué suceder en todos los casos particulares del problema del control del inventario, por lo que se debe determinar al plantear el problema si estas funciones son adecuadas para comparar el ren-

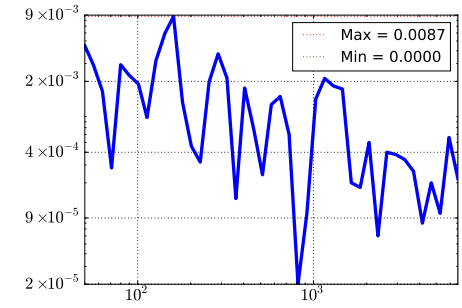
dimiento de los modelos.

Puesto que se ha determinado que *GPR* es un modelo adecuado para afrontar el problema de la predicción de la distribución, sería interesante, de cara al futuro, investigar y probar más configuraciones de las que ofrece el marco teórico de *GPR* en miras de mejorar los resultados obtenidos .
Algunos cambios de configuración que pueden llevarse a cabo son, por ejemplo, la extracción de información a priori de los datos para poder incluirla en el modelo, probar otras funciones de covarianza, funciones de media que no sean cero, incluir distribuciones de probabilidad para los posibles valores de los hiperparámetros, optimizar el modelo marginalizando los parámetros en lugar de aplicar el método de maximización de la verosimilitud marginal, etc.
Cabe destacar que las posibles configuraciones del modelo son ilimitadas y por tanto queda pendiente la realización de muchos más experimentos. Asimismo, existe una diversidad de modelos bayesianos como las *redes bayesianas* o los *modelos ocultos de Markov* que podrían ser también evaluados y comparados con los procesos gaussianos.

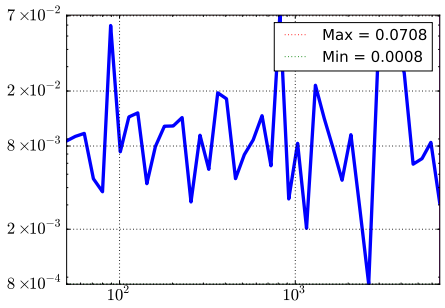
Además, se podría investigar con más detalle la existencia de otras métricas para medir el rendimiento de un modelo en la estimación de cuantiles. En concreto, podría utilizarse alguna métrica que no presentase el planteamiento de EPM y ROFIC de perjudicar las estimaciones infravaloradas, y ver si los resultados permiten otra interpretación sobre qué modelo es más adecuado.
Finalmente los resultados del trabajo parecen indicar que el enfoque bayesiano para resolver el problema de predicción de la distribución para el problema del control del inventario proporciona ventajas considerables frente a los enfoques clásicos de aprendizaje automático.

A Ejemplos de rendimiento en estimación de cuantiles con datos artificiales

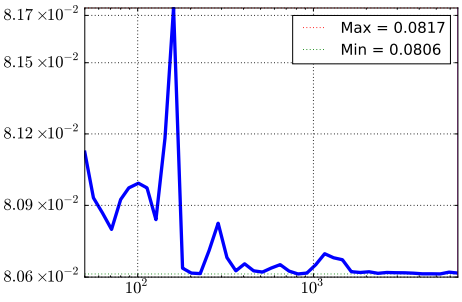
En este anexo pueden encontrarse una serie de gráficas que muestran el rendimiento de los modelos (métricas EPM, EDEC y ROFIC) en la estimación de cuantiles.



(a) ROFIC

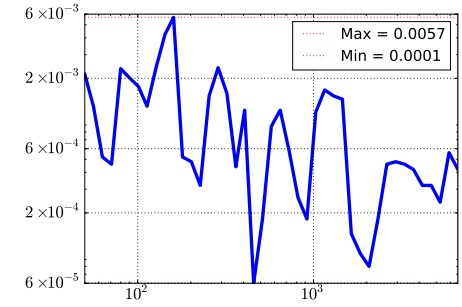


(b) EDEC

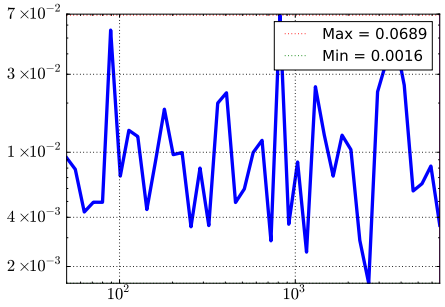


(c) EPM

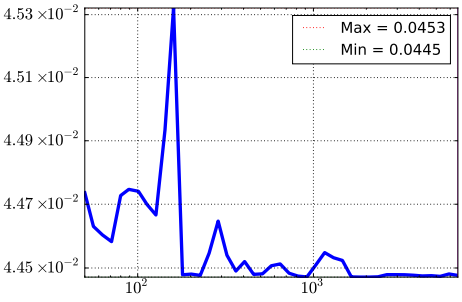
Figura A.1: GPR - Medidas de error frente al tamaño del conjunto de entrenamiento en la estimación del 98-cuantil.



(a) ROFIC

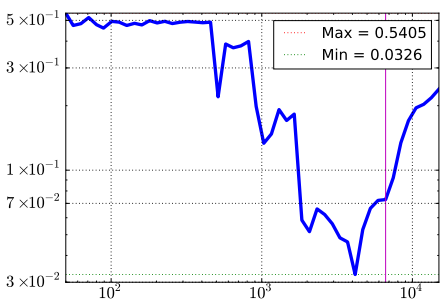


(b) EDEC

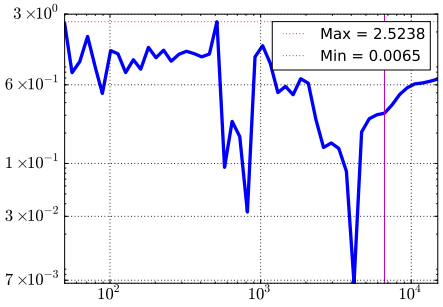


(c) EPM

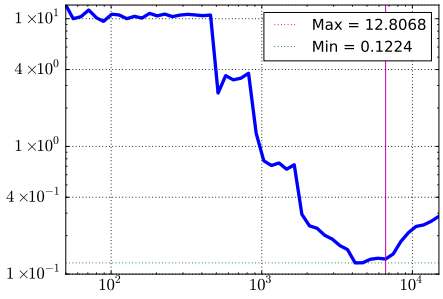
Figura A.2: GPR - Medidas de error frente al tamaño del conjunto de entrenamiento en la estimación del 99-cuantil.



(a) ROFIC

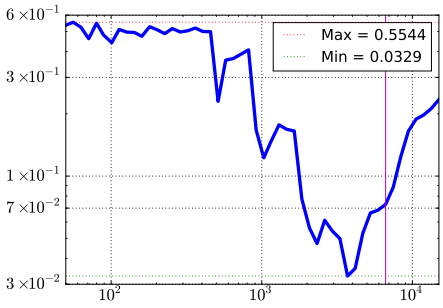


(b) EDEC

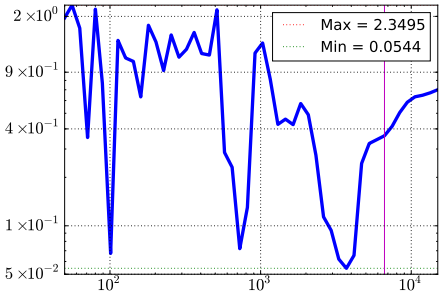


(c) EPM

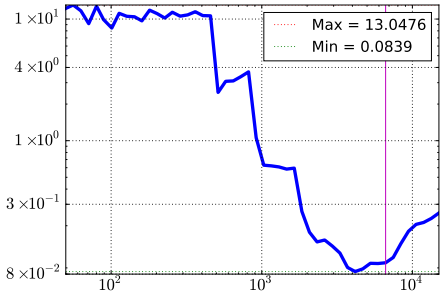
Figura A.3: ABR - Medidas de error frente al tamaño del conjunto de entrenamiento en la estimación del 98-cuantil.



(a) ROFIC

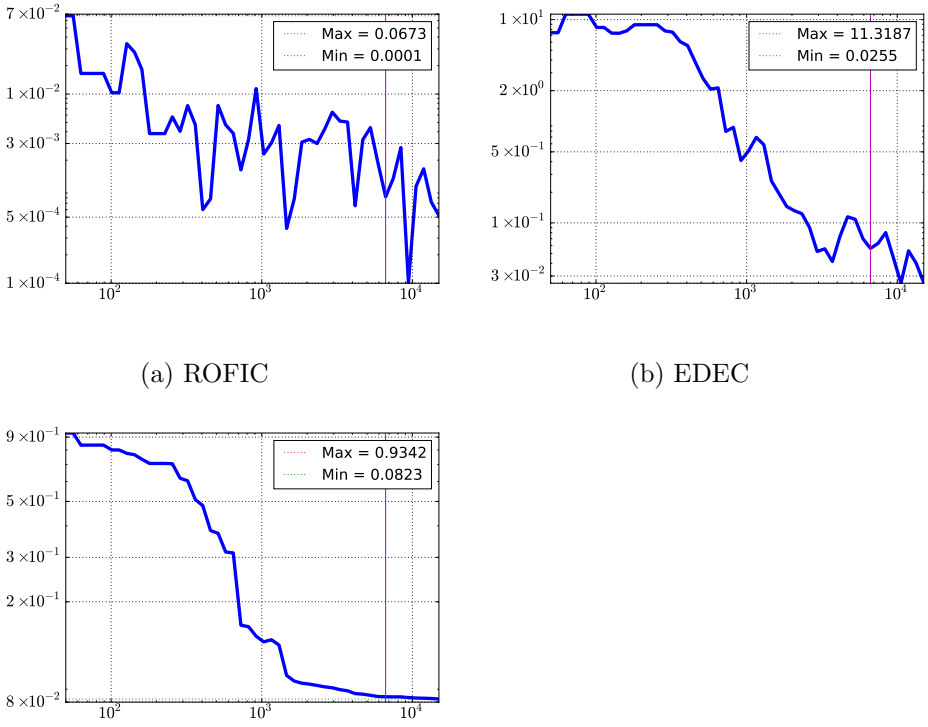


(b) EDEC



(c) EPM

Figura A.4: ABR - Medidas de error frente al tamaño del conjunto de entrenamiento en la estimación del 99-cuantil.

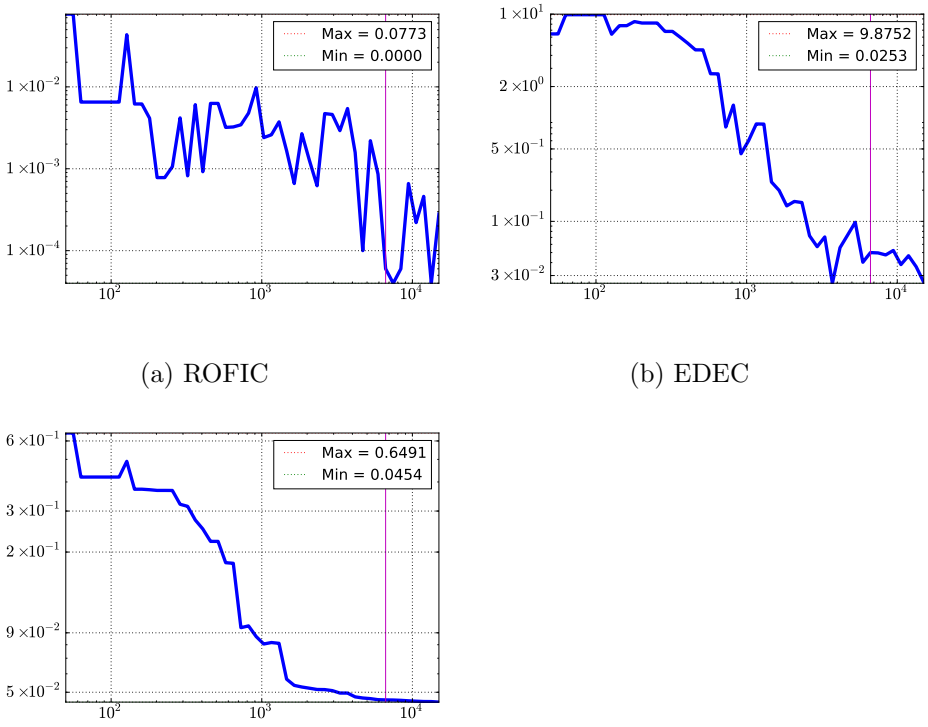


(a) ROFIC

(b) EDEC

(c) EPM

Figura A.5: SVR - Medidas de error frente al tamaño del conjunto de entrenamiento en la estimación del 98-cuantil.



(a) ROFIC

(b) EDEC

(c) EPM

Figura A.6: SVR - Medidas de error frente al tamaño del conjunto de entrenamiento en la estimación del 99-cuantil.

B Rendimiento de los modelos en los experimentos con datos de sucursales

En este anexo se muestran una serie de tablas con datos obtenidos tras la realización de los experimentos en sucursales (Capítulo 4).

En el Cuadro B.1 encontramos los valores de R^2 y EAM que miden la calidad de las predicciones de la demanda para cada sucursal.

En los Cuadros B.2 y B.3 se muestran, respectivamente, los valores de las funciones de error EPM y $ROFIC$, que miden la calidad en la estimación de los cuantiles de la demanda.

ID	R^2			EAM		
	ABR	SVR	GPR	ABR	SVR	GPR
1	0.00621	0.05512	-0.01822	0.02272	0.02278	0.02322
2	-0.01003	-0.16861	-0.01406	0.02114	0.02318	0.02114
3	0.19587	0.13239	0.16600	0.03859	0.04056	0.03851
4	-0.00251	0.03106	0.02865	0.07023	0.06649	0.06807
5	-0.02634	-0.15771	-0.06618	0.04503	0.04825	0.04584
6	0.02010	0.01250	0.01770	0.10845	0.10777	0.10833
7	0.04759	0.01017	0.03636	0.04314	0.04514	0.04280
8	0.14092	0.03691	0.06258	0.05481	0.05800	0.05753
9	0.00204	-0.05081	-0.01946	0.01677	0.01798	0.01724
10	0.16976	-0.29565	0.17207	0.04847	0.06322	0.04757
11	0.13009	0.02835	0.18200	0.05631	0.06188	0.05489
12	0.03464	0.03823	-0.03325	0.07329	0.07269	0.07698
13	0.08944	0.01700	0.08312	0.08643	0.08927	0.08574
14	0.01772	-0.00268	0.01681	0.04105	0.04104	0.04085
15	0.00411	0.00971	-0.00002	0.04267	0.04273	0.04274
16	-0.02372	0.17323	-0.04438	0.07220	0.07169	0.07080
17	-0.00755	-0.02025	-0.01614	0.06534	0.06598	0.06529
18	0.05650	-0.00130	0.06865	0.07909	0.08199	0.07938
19	0.06077	0.04243	-0.00836	0.06516	0.06561	0.06771
20	0.09324	0.07352	0.04506	0.14118	0.14264	0.14493
21	0.03284	0.00285	0.03234	0.08639	0.08769	0.08719
22	0.14983	0.00984	0.17869	0.05037	0.05534	0.04961
23	0.11311	0.07888	0.11016	0.10621	0.11017	0.10553
24	-0.02572	-0.00491	0.00400	0.06417	0.06421	0.06296
25	0.30633	-0.11194	0.35029	0.07047	0.09727	0.06793
26	0.00843	0.01929	-0.00351	0.04968	0.04955	0.04959
27	0.05234	0.03231	0.08322	0.06244	0.06254	0.06216
28	0.07215	0.00691	-0.06284	0.00505	0.00539	0.00578
29	0.01670	-0.00438	-0.00773	0.06058	0.06057	0.06109
30	0.29489	-0.07383	0.32524	0.01469	0.01860	0.01442

Continúa en la siguiente página

Continuación de la página anterior						
31	0.01454	0.00729	0.01900	0.05256	0.05445	0.05189
32	0.11276	0.06514	0.11538	0.04146	0.04276	0.04121
33	0.09728	0.01068	0.03449	0.09051	0.09397	0.09169
34	-1.00393	-1.12980	-0.58735	0.13165	0.13596	0.10043
35	0.31739	0.23372	0.30728	0.05777	0.06293	0.05852
36	0.20820	0.14309	0.09660	0.06981	0.07563	0.07444
37	-0.62493	-0.82464	-0.07283	0.07302	0.08040	0.05649
38	0.23365	0.15931	0.19626	0.03764	0.03882	0.03934
39	0.18419	0.09442	0.22233	0.06831	0.07242	0.06747
40	0.02438	0.07521	0.05831	0.01727	0.01691	0.01707
41	0.48432	0.40777	0.45309	0.04456	0.05104	0.04805
42	-0.13710	-0.18583	-0.12923	0.03048	0.03125	0.02972
43	-0.11092	-0.48431	-0.20240	0.07731	0.09126	0.08071
44	0.74729	0.38670	0.65203	0.03235	0.05332	0.03722
45	0.47777	0.39688	0.41332	0.03086	0.03328	0.03399
46	0.01580	-0.06617	0.01954	0.10749	0.11104	0.10942
Media	0.06653	-0.01722	0.07097	0.05924	0.06273	0.05877

Cuadro B.1: R^2 y EAM obtenidos por cada modelo en las sucursales consideradas.

ID	cuantil 95 %			cuantil 98 %			cuantil 99 %		
	ABR	SVR	GPR	ABR	SVR	GPR	ABR	SVR	GPR
1	0.00731	0.00428	0.00403	0.00690	0.00231	0.00200	0.00677	0.00150	0.00109
2	0.00388	0.00435	0.00399	0.00179	0.00219	0.00196	0.00110	0.00118	0.00110
3	0.00520	0.00520	0.00570	0.00349	0.00252	0.00290	0.00292	0.00160	0.00174
4	0.01501	0.01311	0.01402	0.01290	0.00771	0.00900	0.01219	0.00579	0.00679
5	0.00867	0.00835	0.00865	0.00674	0.00501	0.00499	0.00609	0.00352	0.00336
6	0.02219	0.02005	0.01477	0.01934	0.01191	0.00783	0.01838	0.00778	0.00466
7	0.00843	0.00749	0.00794	0.00454	0.00410	0.00418	0.00325	0.00281	0.00256
8	0.00820	0.00838	0.00816	0.00474	0.00430	0.00418	0.00359	0.00269	0.00251
9	0.00520	0.00328	0.00409	0.00490	0.00181	0.00202	0.00480	0.00105	0.00115
10	0.00868	0.00705	0.00745	0.00376	0.00353	0.00369	0.00212	0.00212	0.00203
11	0.01035	0.01138	0.01004	0.00631	0.00595	0.00528	0.00497	0.00344	0.00316
12	0.01895	0.01312	0.01286	0.01725	0.00703	0.00713	0.01668	0.00413	0.00452
13	0.01317	0.02379	0.01284	0.00795	0.01217	0.00664	0.00620	0.00915	0.00415
14	0.00851	0.00686	0.00689	0.00677	0.00382	0.00377	0.00620	0.00250	0.00246
15	0.01000	0.00661	0.00646	0.00859	0.00310	0.00304	0.00811	0.00181	0.00163
16	0.01391	0.01179	0.01192	0.01173	0.00689	0.00654	0.01101	0.00468	0.00423
17	0.01728	0.01076	0.01106	0.01581	0.00550	0.00620	0.01531	0.00333	0.00411
18	0.01356	0.01509	0.01193	0.00710	0.00772	0.00647	0.00495	0.00482	0.00418
19	0.01618	0.00895	0.00847	0.01494	0.00420	0.00412	0.01452	0.00267	0.00229
20	0.03131	0.02187	0.02509	0.02665	0.01115	0.01400	0.02510	0.00722	0.00918
21	0.02224	0.01938	0.01463	0.02052	0.01109	0.00923	0.01995	0.00660	0.00678
22	0.01038	0.01027	0.00954	0.00813	0.00566	0.00577	0.00737	0.00361	0.00413
23	0.02415	0.01600	0.01573	0.02143	0.00721	0.00844	0.02052	0.00370	0.00523
24	0.01738	0.00787	0.00845	0.01203	0.00438	0.00404	0.01025	0.00306	0.00225
25	0.01311	0.01846	0.01275	0.01023	0.00661	0.00755	0.00926	0.00363	0.00509
26	0.01198	0.00929	0.00937	0.01038	0.00512	0.00523	0.00984	0.00347	0.00329
27	0.01433	0.00933	0.00897	0.01273	0.00440	0.00469	0.01220	0.00247	0.00288
28	0.00146	0.00106	0.00224	0.00135	0.00066	0.00111	0.00131	0.00045	0.00063
29	0.01671	0.00933	0.00960	0.01563	0.00515	0.00516	0.01527	0.00315	0.00318
30	0.00185	0.00208	0.00215	0.00096	0.00097	0.00104	0.00066	0.00054	0.00058

Continúa en la siguiente página

Continuación de la página anterior									
31	0.01000	0.00702	0.00828	0.00481	0.00412	0.00413	0.00308	0.00343	0.00234
32	0.01039	0.00754	0.00718	0.00551	0.00418	0.00398	0.00389	0.00281	0.00267
33	0.01360	0.01501	0.01524	0.00798	0.00742	0.00887	0.00611	0.00485	0.00605
34	0.05885	0.03286	0.02165	0.05982	0.01291	0.01209	0.06014	0.00658	0.00802
35	0.01041	0.00873	0.00946	0.00750	0.00407	0.00489	0.00654	0.00234	0.00298
36	0.01707	0.01357	0.01372	0.01473	0.00707	0.00764	0.01395	0.00450	0.00492
37	0.04598	0.01586	0.00803	0.04703	0.00483	0.00401	0.04738	0.00329	0.00221
38	0.00668	0.00682	0.00788	0.00427	0.00352	0.00373	0.00346	0.00220	0.00209
39	0.01504	0.00957	0.00826	0.01338	0.00448	0.00398	0.01283	0.00269	0.00215
40	0.00345	0.00208	0.00396	0.00301	0.00144	0.00195	0.00286	0.00083	0.00110
41	0.00896	0.00872	0.00758	0.00710	0.00453	0.00434	0.00648	0.00281	0.00293
42	0.01799	0.00643	0.00507	0.01824	0.00424	0.00282	0.01832	0.00316	0.00186
43	0.01025	0.01214	0.01097	0.00640	0.00565	0.00528	0.00512	0.00299	0.00294
44	0.00577	0.00677	0.00558	0.00263	0.00389	0.00269	0.00158	0.00262	0.00155
45	0.01027	0.00470	0.00480	0.00537	0.00249	0.00238	0.00373	0.00147	0.00139
46	0.02307	0.01401	0.01301	0.02060	0.00576	0.00566	0.01978	0.00276	0.00316
Media	0.01407	0.01058	0.00958	0.01161	0.00532	0.00514	0.01079	0.00334	0.00325

Cuadro B.2: EPM obtenido por cada modelo para los cuantiles 95 %, 98 % y 99 % en las 46 sucursales estudiadas.

ID	cuantil 95 %			cuantil 98 %			cuantil 99 %		
	ABR	SVR	GPR	ABR	SVR	GPR	ABR	SVR	GPR
1	0.21087	0.03075	0.01273	0.24087	0.02348	0.00174	0.25087	0.02727	0.00863
2	0.03137	0.04379	0.04379	0.00137	0.02000	0.01689	0.00863	0.01000	0.01000
3	0.04006	0.05559	0.03447	0.07006	0.00137	0.01379	0.08006	0.00689	0.00689
4	0.10528	0.04006	0.06801	0.13528	0.02037	0.06075	0.14528	0.02106	0.04280
5	0.05248	0.00652	0.02516	0.08248	0.00795	0.00447	0.09248	0.01174	0.00553
6	0.21087	0.16118	0.00901	0.24087	0.12596	0.02037	0.25087	0.09870	0.02106
7	0.01273	0.00590	0.01584	0.01727	0.01379	0.00174	0.02727	0.00689	0.00553
8	0.00590	0.01211	0.00342	0.03590	0.00484	0.00484	0.04590	0.00379	0.00553
9	0.22640	0.03137	0.04068	0.25640	0.01068	0.01379	0.26640	0.00068	0.00689
10	0.03095	0.01508	0.00397	0.00095	0.01048	0.01810	0.00905	0.01000	0.01540
11	0.00031	0.03075	0.00031	0.02969	0.00137	0.00484	0.03969	0.00553	0.00553
12	0.19534	0.02205	0.00963	0.22534	0.00137	0.00174	0.23534	0.00242	0.00863
13	0.00590	0.17360	0.00652	0.03590	0.10422	0.00795	0.04590	0.11112	0.00242
14	0.09286	0.00590	0.01584	0.12286	0.00447	0.00447	0.13286	0.00068	0.00068
15	0.13634	0.02453	0.01273	0.16634	0.00758	0.00174	0.17634	0.00689	0.00379
16	0.13323	0.03696	0.02826	0.16323	0.01727	0.00758	0.17323	0.01484	0.00242
17	0.17981	0.01273	0.02764	0.20981	0.00447	0.02658	0.21981	0.00379	0.02416
18	0.00031	0.08354	0.00280	0.02969	0.02037	0.01416	0.03969	0.02727	0.01174
19	0.25745	0.05559	0.02205	0.28745	0.01379	0.01068	0.29745	0.00689	0.00689
20	0.13095	0.01667	0.06429	0.16095	0.01492	0.05937	0.17095	0.00905	0.04714
21	0.26366	0.21087	0.04627	0.29366	0.06385	0.03901	0.30366	0.02106	0.03658
22	0.05248	0.03075	0.00652	0.08248	0.00795	0.00484	0.09248	0.00553	0.01174
23	0.18292	0.04938	0.02143	0.21292	0.01416	0.02658	0.22292	0.00242	0.03037
24	0.12391	0.03447	0.03137	0.15391	0.02000	0.01689	0.16391	0.01000	0.00689
25	0.04627	0.15186	0.01832	0.07627	0.01106	0.02969	0.08627	0.01484	0.03037
26	0.10839	0.04317	0.01273	0.13839	0.00484	0.00795	0.14839	0.01484	0.01174
27	0.17981	0.01211	0.02453	0.20981	0.00447	0.02969	0.21981	0.00553	0.02106
28	0.18602	0.05559	0.04689	0.21602	0.02969	0.02000	0.22602	0.01174	0.01000
29	0.28230	0.01584	0.02516	0.31230	0.00447	0.00174	0.32230	0.00379	0.00553
30	0.00031	0.01211	0.03447	0.02969	0.01416	0.01689	0.03969	0.01484	0.00689

Continúa en la siguiente página

Continuación de la página anterior									
31	0.00652	0.00963	0.03447	0.02348	0.01379	0.01379	0.03348	0.01000	0.00379
32	0.00031	0.02143	0.01894	0.02969	0.00447	0.01068	0.03969	0.00242	0.00379
33	0.00590	0.03696	0.04938	0.03590	0.01106	0.02969	0.04590	0.01795	0.03969
34	0.59596	0.38789	0.14565	0.62596	0.12596	0.09491	0.63596	0.05211	0.06453
35	0.04317	0.01522	0.00652	0.07317	0.00484	0.00484	0.08317	0.00242	0.00068
36	0.16118	0.01584	0.00963	0.19118	0.00758	0.01727	0.20118	0.00689	0.01484
37	0.70155	0.31335	0.02205	0.73155	0.03590	0.00174	0.74155	0.02106	0.00068
38	0.01522	0.00963	0.03447	0.04522	0.01068	0.01379	0.05522	0.00689	0.01000
39	0.21087	0.04006	0.02826	0.24087	0.00137	0.01068	0.25087	0.00689	0.00379
40	0.21398	0.02826	0.05000	0.24398	0.02000	0.02000	0.25398	0.01000	0.01000
41	0.09286	0.06801	0.00280	0.12286	0.02037	0.01106	0.13286	0.00242	0.01484
42	0.52143	0.10217	0.01832	0.55143	0.08248	0.00795	0.56143	0.08006	0.01174
43	0.01832	0.03447	0.02516	0.04832	0.01068	0.00758	0.05832	0.00068	0.00242
44	0.02205	0.03696	0.02826	0.00795	0.02000	0.01379	0.01795	0.01000	0.00689
45	0.04627	0.01211	0.00963	0.07627	0.00758	0.00137	0.08627	0.00068	0.00242
46	0.22019	0.02143	0.00280	0.25019	0.02658	0.01068	0.26019	0.01000	0.01000
Media	0.13829	0.05727	0.02611	0.16383	0.02189	0.01649	0.17373	0.01588	0.01333

Cuadro B.3: ROFIC obtenido por cada modelo para los cuantiles 95 %, 98 % y 99 % en las 46 sucursales estudiadas.

Bibliografía

[1] E. L. Porteus. *Foundations of Stochastic Inventory Theory*. Stanford University Press, 2002.

[2] R.O. Duda, P.E. Hart y D.G. Stork. *Pattern classification*. Wiley New York, 2001.

[3] I. Takeuchi y col. «Nonparametric Quantile Regression». En: *Journal of Machine Learning Research* 7(4) (2006), págs. 1231-1264.

[4] W.T. Shaw y G. Steinbrecher. «Quantile mechanics». En: *European Journal of Applied Mathematics* 19 (2) (2008), págs. 87-112.

[5] C. E. Rasmussen y C. K. I. Williams. *Gaussian Processes for Machine Learning*. Ed. por MIT Press. MIT Press, 2006.

[6] Y. Freund y R.E. Schapire. «A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting». En: *Journal of Computer and System Sciences.* 55 (1997), pág. 119 139.

[7] H. Drucker. «Improving Regressors using Boosting Techniques». En: *In Douglas H. Fisher, ed., ICML, Morgan Kauffman 107-115*. 1997.

[8] T. Hastie, R. Tibshirani y J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Ed. por Statistics. Springer, 2001.

[9] L. Breiman y col. *Classification and Regression Trees*. Ed. por Cole Advanced Books & Software. Wadsworth & Brooks, 1984.

[10] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[11] C. Cortes y V. Vapnik. «Support-Vector Networks». En: *Machine Learning* 20 (1995), págs. 1-25.

[12] V. Vapnik. *The Nature of Statistical Learning Theory*. Ed. por Verlag. Springer, 1995.

[13] B. Schölkopf y A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[14] H. Drucker y col. *Support Vector Regression Machines*. Vol. 9. Advances in Neural Information Processing Systems. MIT Press, 1997, págs. 155-161.

[15] I. Steinwart y A. Christmann. «Estimating Conditional Quantiles With the Help of the Pinball Loss». En: *Bernoulli* 17(1) (2011), págs. 211-225.

[16] A. Boukouvalas, R. Barillec y D. Cornford. «Direct Gaussian Process Quantile Regression using Expectation Propagation». En: *Proceedings of the 29 th International Conference on Machine Learning*. Edinburgh, Scotland, jun. de 2012.

[17] Jarno Vanhatalo y col. «GPstuff: Bayesian Modeling with Gaussian Processes». En: *Journal of Machine Learning Research* 14 (abr. de 2013), págs. 1175-1179.